AFOSR-TR-97-0714

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1, 1997	Final Report - February 1, 1994 to June 14, 1997

4. TITLE AND SUBTITLE
(AASERT 94–045) Holographic Associative Memories for Database Processing

6. FUNDING NUMBERS
F49620–94–1–0361

6. AUTHOR(S)
Dr. Pericles A. Mitkas

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523-1373

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
AFOSR/PKA
110 Duncan Ave., Suite B115
Bolling AFB, DC 20332-0001

NE

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
Final Report

11. SUPPLEMENTARY NOTES

a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

19971217 077

13. ABSTRACT *(Maximum 200 words)*

Volume holographic memories combine high storage capacity, massive transfer rates, and the ability to associatively search the entire data set in a highly parallel way. Database applications can benefit greatly from these characteristics. The goal of this research was to evaluate the capability of volume holographic storage to perform associative recall on binary data of an actual database and, thus, operate successfully in a database management environment. An angularly multiplexed volume holographic database system (VHDS) was developed and used for storing and searching alphanumeric database records and image data. VHDS operates in three modes: a) the recording mode, b) the addressed–based retrieval mode, and c) the content–based retrieval mode. In the latter mode, the $LiNbO_3$ crystal is illuminated by a search pattern and a reference beam profile is reconstructed with peaks pointing to the addresses of pages that contain the search pattern. These pages can be retrieved later with the system operating in the addressed–based retrieval mode. VHDS has operated in all three modes. Several hundreds of pages of binary data have been recorded and good associative recall with search arguments as small as 1/30th of a page has been demonstrated. A first order analysis of the factors that affect the quality of the reconstructed reference beam profile has been completed. Both digital and analog data can be combined in the same recording and searching is possible with either data type. This functionality can be very useful in multimedia database management systems and a whole array of applications that use different types of data.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Volume Holographic Memories, Volume Holographic Database System, Content-based Retrieval, Address-based Retrieval, Analog and Digital Data	56
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

HOLOGRAPHIC ASSOCIATIVE MEMORIES FOR DATABASE PROCESSING

Grant #F49620–94–1–0361

Final Technical Report

Submitted by
Dr. Pericles A. Mitkas, PI

Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523–1373

November 1997

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Volume holographic memories combine high storage capacity, massive transfer rates due to page–oriented access, and the ability to associatively search the entire data set in a highly parallel way. Database applications may benefit greatly from these characteristics of holographic memories. A significant percentage of existing databases contain strictly alphanumeric data. With the advent of multimedia, databases are expected to accommodate non-alphanumeric information such as images and sound as well. The most popular technique for managing a database is the storage of data in records comprising several well-defined data fields that are arranged in a prespecified and strict format. A number of database operations are defined that include searching for a particular record based on a data attribute or unique identifier and retrieval of a subset of records that satisfy a user supplied selection criterion.

The goal of our research was to determine the capability of volume holographic storage to perform associative recall on binary data of an actual database and, thus, operate successfully in a database management environment.

The main objectives were:

a) Development of a scheme for recording a relational database in holographic memories in a way that allows data access based on either physical address or content.

b) Design and analysis of an optoelectronic system based on the above scheme.

c) Implementation of a prototype system to evaluate the ability of holographic storage to associatively search binary encoded data.

This research was initiated in February of 1994 and ended in August of 1997. It was supported by a parent AFOSR project (#F49620–94–1–0148) and an AASERT project (#F49620–94–1–0361).

We have developed an angularly multiplexed volume holographic associative memory for storing and searching alphanumeric database records and image data. The system, called VHDS, uses iron-doped lithium niobate as the storage medium and operates in three modes: a) the recording mode, when data pages are loaded on the SLM and written in the crystal, b) the addressed–based retrieval mode, when a certain page is chosen and retrieved with the reference beam incident on the crystal at a given angle (address), and c) the content–based retrieval mode. In the latter mode, the crystal is illuminated by a search pattern loaded on the SLM and a CCD camera receives the reconstructed reference beam profile that has peaks pointing to the addresses of pages that contain the search pattern. These pages can be retrieved later with the system operating in the addressed–based retrieval mode.

We have demonstrated operation of VHDS in all three modes and have automated most of the processes using a PC. We have recorded several hundred of pages of binary data (both random and real) encoded according to a 2–out–of–15 modulation scheme with a multiblock cross–parity error correcting code. We have demonstrated good associative recall with search arguments as small as 1/30th of a page and performed a first order analysis of the factors that affect the quality of the reconstructed reference beam profile.

The results show that good associative addressability can be achieved by thresholding the reconstructed reference beam profiles. More elaborate signal processing techniques can be used to further reduce the percentage of false hits. We have also shown that such a memory provides position, orientation, and bit size variance which is desirable when searching relational data-

bases. Both digital and analog data can be combined in the same recording and searching can be performed on both types of data. This functionality can be very useful in multimedia database management systems and a whole array of applications that use different types of data. In performing experiments that combined both analog and digital data we were able to observe that the intensity of hits in the reference plane is dependent not only upon search argument size, but also content or the stored data's diffraction efficiency. This means that steps need to be taken to ensure that both the digital and analog data pages are recorded at the same efficiency. Mixing alphanumeric and analog data provided these benefits: a) wider angular separation, b) greater intensity of reconstructed reference beams when searching for analog data, c) lower cross–correlation due to mixed data types and fewer binary data pages, and d) the ability to remove some false hits that corresponded to analog page positions when digital search arguments were used and vice versa.

Future work should include a) full automation of the system, b) further characterization of the factors affecting the quality of associative recalls and more specifically the effect of the encoding scheme, c) memory improvements and more specifically attempts to increase the memory's capacity, to reduce the BER, and to improve the contrast between hits and non-hits in the reference plane. Capacity increase can be accomplished by improving the optics' quality to enhance the system's resolution, which may also lead to a lower BER. Capacity can also increase through the use of a better quality SLM and the use of spatial multiplexing.

Extensions of the system's functionality may include video indexing, image compression techniques, and interactive video applications.

DEPOM '96

Another major accomplishment during the course of this project was the organization of the First Workshop on Data Encoding for Page-oriented Optical Memories (DEPOM'96) in Phoenix, AZ, March 27–28, 1996. The Workshop was sponsored by the Rome Laboratory, the Air Force Office of Scientific Research, and Colorado State University. The purpose of DEPOM was to bring together researchers in the following areas: Page-oriented optical memory (POM) systems; Array codes for error control in POMs; Error generating processes in POMs; Channel characterization and modelling; Data encoding schemes for POMs; Hardware schemes for parallel error control and decoding in POMs. DEPOM'96 was attended by 53 people representing 14 Universities, 10 companies, and 3 Government Agencies. The Workshop Proceedings were published in a volume.

Related Research Projects

1.. Holographic Search Engine for Multimedia Databases

The major objective of this one year project funded by **DARPA** will be to develop a holographic search engine based on the Texas Instruments Digital Micromirror Display. The system will be an extension of VHDS. Replacing the liquid crystal display we are currently using with this high quality SLM is expected to increase the storage capacity of the system, reduce the noise level during associative recall, and significantly improve the overall system performance. We plan to combine different types of data and attempt content–based searches. These improvements are expected to enable the development and demonstration of a practical system.

2.. Reference Beam Reconstruction during Associative Recall in Digital Holographic Memories

In a collaborative research project funded by **NATO** and currently under way, we work together with the Institute of Laser and Material Structure of the Foundation of Research and Technology in Greece in order to characterize the reference beam profile of VHDS using different materials (SBN, KTO) and different laser sources.

3.. Holographic Search Engine for Multimedia Databases

This one year project is funded by the **NSF Optoelectronic Computing Systems Center** and aims to develop a holographic search engine for various types of data. Such a special–purpose optoelectronic processing unit can be added as a peripheral to an electronic host and undertake computationally intensive operations (mainly content–based searches) in a multimedia environment. In order to satisfy this objective we have to characterize the response of VHDS for different types of data. We have chosen four applications each of which uses a different data type:

a) Database management (digital alphanumeric data)
b) Video indexing (gray scale and color images)
c) Image compression by vector quantization (binary images)
d) Interactive video (combinations of alphanumeric data, binary and gray scale images).

4.. Error Detection and Correction Codes for Optical Memories with 2D Parallel Output

This 18–month project was funded by the **Air Force Rome Laboratory** and its main objectives were: a) to investigate the error control process during data readout in optical memories with 2D parallel output and b) develop and evaluate encoding schemes for efficient error detection and correction. A large number of array codes were evaluated and compared and a software CAD tool was developed to assist in the analysis of the interface between page–oriented optical memories and electronic computers. Experimental results from VHDS were used repeatedly in this project.

5.. Volume Holographic Storage for Digital Data

This two-year project is funded by the **Storage Technology Corporation** and the **Colorado Advanced Technology Institute**. The purpose of this research is to investigate the feasibility of a volume holographic memory system for storing digital data. We have focused our efforts on the development of data recording and read back techniques required for future holographic storage devices. We are investigating various 3D data encoding schemes. VHDS, again, provides the test bed for the generation of experimental data.

Personnel

The parent project supported only the principal investigator, Dr. Pericles A. Mitkas. The AASERT supported the following graduate students: Bradley J. Goertzen, Keith G. Richling, Michael Porter, and Terry Garrett. The first two were supported throughout their graduate studies at Colorado State University by this project.

Theses and Publications

The following Theses and publications were generated.

Theses

6.. B. J. Goertzen, Volume Holographic Storage for Large Relational Databases, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, 1995.

7.. K. G. Richling, Characterization of Associative Data Recall in Volume Holographic Memories, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, 1997.

8.. M. Porter, Data Recovery Schemes for Page–oriented Optical Memories, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, to be defended in 1998.

Journal Publications

9.. K. G. Richling, B. J. Goertzen, and P. A. Mitkas, "Holographic Associative Processing for Large Databases," under revision, *Applied Optics*, 1998.

10.. L. J. Irakliotis, C. W. Wilmsen, and P. A. Mitkas, "The Optical Memory/Electronic Computer Interface as a Parallel Processing Architecture," *J. of Parallel and Distributed Computing*, Vol. 41, 1997, pp. 67-77.

11.. L. J. Irakliotis and P. A. Mitkas, "Three-dimensional Tomographic Storage Scheme for Optoelectronic Implementation of Database Operations," *Journal of Optical Memories and Neural Networks,* Vol. 5, No. 3, 1996, pp. 155–169.

12.. B. J. Goertzen, K. G. Richling, and P. A. Mitkas, "Implementation of a Volume Holographic Database System," *Optical Review*, Vol. 3, No. 6A, Nov./Dec. 1996, pp. 385–387

13.. B. J. Goertzen and P. A. Mitkas, "Volume Holographic Storage for Large Relational Databases," *Optical Engineering*, Vol. 35, No. 7, July 1996, pp. 1847–1853.

14.. B. J. Goertzen and P. A. Mitkas, "An Error Correcting Code for Volume Holographic Storage of a Relational Database," *Optics Letters*, Vol. 20, No. 15, Aug. 1, 1995, pp. 1655-1657.

15.. P. A. Mitkas and L. J. Irakliotis, "Three–dimensional Optical Storage for Database Processing," *Journal of Optical Memories and Neural Networks,* Vol. 3, No. 2, 1994, pp. 217–229.

Book Chapter

16.. P. A. Mitkas, L. J. Irakliotis, and G. Betzos, "Optical Associative Processors," invited book chapter in Associative Processing and Processors, edited by A. Krikelis and C. Wheems, IEEE Computer Society Press, June 1997.

Conference Proceedings

17.. Proceedings of the 1996 Workshop on Data Encoding for Page-oriented Optical Memories (DEPOM '96), P. A. Mitkas, Editor, Colorado State University College of Engineering, 1997.

Conference Presentations

18.. P. A. Mitkas, G. Betzos, S. Mailis, and N. Vainos, "Characterization of Associative Recall in a Volume Holographic Database System for Multimedia Applications," to be presented at AeroSense '98, Advances in Optical Information Processing, Orlando, FL, April 13–17, 1998.

19.. K. G. Richling, P. A. Mitkas, S. Mailis, L. Boutsikaris, and N. Vainos, "Holographic Storage and Associative Processing of Analog and Digital Data," Proceedings of the IEEE LEOS 10th Annual Meeting, San Fransisco, CA, November 10–13, 1997, Vol. 1, pp. 132-133.

20.. G. A. Betzos, J. F. Hutton, M. S. Porter, and P. A. Mitkas, "Error Correction for Page-oriented Memories," presented at the OSA Annual Meeting, Long Beach, CA, October 12–17, 1997.

21.. P. A. Mitkas, "Holographic Storage for Large Database Applications," presented at the International Symposium on Holographic Memories '96, Vouliagmeni, Greece, May 12–15, 1996.

22.. K. G. Richling and P. A. Mitkas, "Characterization of Volume Holographic Associative Searching," presented at the OSA Holography Topical Meeting, Boston, MA, April 28–30, 1996.

23.. B. J. Goertzen, K. G. Richling, and P. A. Mitkas, "Implementation of a Volume Holographic Database System," presented at the International Topical Meeting on Optical Computing, OC '96, April 21-25, 1996, Sendai, Japan.

24.. B. J. Goertzen and P. A. Mitkas, "Reliable Storage and Retrieval of Database Records using a Volume Holographic Medium," presented at the 1995 OSA Annual Meeting, Portland, OR, September 10–15, 1995.

25.. P. A. Mitkas, "Parallel Optoelectronic Storage for Image Databases," presented at the 1995 International Workshop on Stereoscopic and Three-dimensional Imaging (IWS3DI'95), Santorini, Greece, September 6–8, 1995.

26.. L. Irakliotis and P. A. Mitkas, "Tomographic Storage Scheme for 3D Implementation of Database Operations," presented at the 39th SPIE International Symposium on Optical Applied Science and Engineering, Conference on Very Large Optical Memories, San Diego, CA, July 1994. Paper appears in *Photonics for Processors, Neural Networks and Memories II*, J. L. Horner, B. Javidi, and S. T. Kowell (eds), Proceedings of the SPIE, Vol. 2297, pp. 359–367, 1994.

27.. J. Brown, L. J. Irakliotis, and P. A. Mitkas, "Volume Holographic Storage and Processing for Large Databases," Proceedings of the OSA Topical Meeting on Optical Computing, Vol. 7, Palm Springs, CA, March 1993, pp. 54–57.

1. Problem Statement and Objectives of this Research

Computing applications continually put increasing demands on their secondary storage systems, prompting a great deal of research into optical solutions which may satisfy these demands. Commercially available optical memories, mainly in the form of optical disks, offer attractive capacities and data densities. Emerging optical technologies such as holographic memories and other three-dimensional approaches are expected to deliver even higher capacities in addition to massive transfer rates due to the ability to readout data in a 2-D format [BUR94, HON95, LI94, MAN91, MOE95, MOK93, PSA95a, PSA95b, RED92, WUL94].

One of the most demanding applications on secondary storage is the management of large databases. Database systems differ from many other computing applications in that they are input/output intensive rather than computationally intensive. Thus, their performance depends more heavily on the capabilities of the I/O channel than of the CPU [BER89].

Database applications may benefit greatly from the potentially high storage capacity and parallelism of holographic memories. But these memories exhibit additional characteristics that are desirable in database storage systems. First, holographic memories have a page-oriented data format which dovetails nicely with the tabular format of relational database records. Second, holographic memories can be quickly accessed based on content, rather than physical address. Indeed, holographic associative memories have been demonstrated repeatedly in the past, but, to the best of our knowledge, all these systems have used analog images recorded as grey-scale data. The goal of our research was to determine the capability of volume holographic storage to perform associative recall on binary data of an actual database and, thus, operate successfully in a database management environment.

The main objectives were:

a) Development of a scheme for recording a relational database in holographic memories in a way that allows data access based on either physical address or content.

b) Design and analysis of an optoelectronic system based on the above scheme.

c) Implementation of a prototype system to evaluate the ability of holographic storage to associatively search binary encoded data.

2. Volume Holographic Storage for Large Databases

A significant percentage of existing databases contain strictly alphanumeric data. Examples include the databases of financial institutions, personnel files, inventories, etc. With the advent of multimedia, databases are expected to accommodate non-alphanumeric information such as images and sound as well. Techniques can be employed to convert binary data to a strictly alphanumeric form. This conversion increases the size of the binary file, but allows binary data to be easily incorporated in alphanumeric databases. The most popular technique for managing a database is the arrangement of data in records comprising several well-defined data fields. Thus, the data are stored in a prespecified and strict order which facilitates the execution of a number of database operations on them. Typical operations include searching for a particular record based on a data attribute or unique identifier and retrieval of a subset of records that satisfy a user supplied selection criterion.

An example of a database formatted as described above is shown in Figure 1. The table lists the mountain bike models available at a local bike shop. Each row in the table contains one

record while each column delineates an attribute of the record. A query through the database for those bikes having front suspension only shows that the shop has two such bikes (shaded in the figure). Such a query (called a selection operation in a relational database environment) can be executed in many different ways. The most straightforward, but also extremely slow, method is based on serially retrieving each record and checking the entry in its "Suspension" field against the search argument "Front." The best selection protocol would search the entire database associatively in a single step and retrieve only the qualifying records.

Manufacturer	Model	Suspension	Suggested Retail, $
Cannondale	M–400	None	615
Cannondale	F–1000	Front	1675
Univega	Alpina 507	None	769
Univega	Aluminum 708–X	Front	1000
Barracuda	A2Fast	Front and Rear	1299
Barracuda	A2M	None	859
GT	Karakoram	None	750

Figure 1. Example of several database records. The shaded rows show the results of a search for bikes with front suspension only.

Holographic memories lend themselves nicely to the storage and retrieval of relational databases because these memories have large theoretical storage capacities, are page-oriented, and can be associatively searched. Holographic memories have theoretical storage capacities that are greater than 1 TB/cm^3 and can provide a mechanism to store vast amounts of information in a relatively small volume. The page-oriented nature of holographic memories also maps well to the tabular format of relational databases because each row, or record, can be modulated and stored as a single page. If the modulation scheme keeps attributes in the same place on each page, then the memory can be associatively searched. Holographic memories are a good choice for storing multimedia databases, as well, because both analog images and digital data can be combined in the same recording and searched associatively.

Associative searching is highly desirable because it reduces the amount of data that must be transported between computers and their secondary storage devices. In conventional searching techniques, a record is read from the memory and a comparison of the desired attributes is performed. If the comparison is successful, the record is kept, otherwise it is discarded and the next one is read. This process continues until the entire contents of the database are exhausted. In contrast, associative searching is performed by providing the memory with the search argument(s). In a single step, the memory generates the addresses of the pages that contain the desired data. We now read only those pages, effectively reducing the I/O traffic in the system.

In summary, the large capacity, page-oriented nature, and associative searching abilities of holographic memories make them a good secondary storage device for both relational and multimedia databases.

2.1. Volume Holographic Database System (VHDS)

A block diagram of the volume holographic database system (VHDS) that we have developed is depicted in Figure 2. The system consists of two arms, each composed of two blocks. The reference beam generation and detection blocks comprise the reference arm, while the image generation and detection blocks form the image arm. We describe the functionality of these blocks briefly here and provide more details about the implementation of the entire system in section 5.

Light from a single laser source is split into the two arms of the system. The image beam generation system uses an electrically addressed spatial light modulator to convert the input light into a page of binary data. The spatial Fourier transform of the page is then recorded in the storage medium as a volume hologram. Recording Fourier holograms has some advantages such as increased misalignment tolerance and better noise immunity because the spatial pattern is converted to a spatial frequency spectrum resulting in distribution of information from each bit throughout the medium. Any disadvantages of recording Fourier holograms, such as interpixel cross talk due to the finite size of the recording medium and non-uniform diffraction efficiencies due to intense DC components, are overshadowed by the main reason behind our choice: Fourier holograms are essential to our application because they facilitate associative searching and retrieval [COL71].

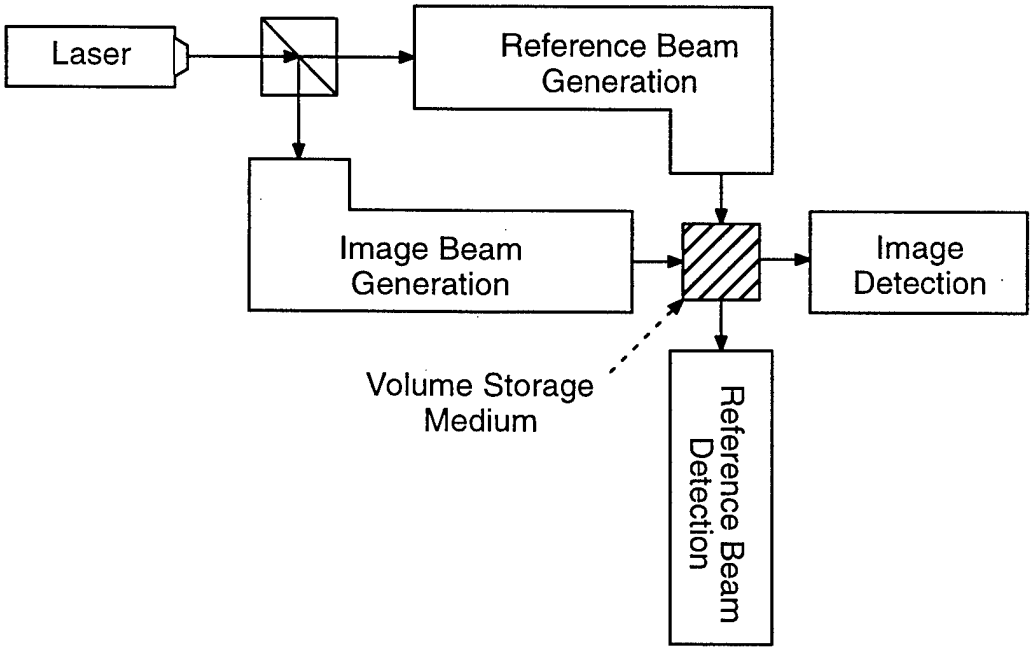


Figure 2. Block diagram of the volume holographic database system.

The reference beam generation section provides a reference beam incident on the medium at a unique angle for each page, which corresponds to that page's address. The interference pattern of the two beams is then stored in the medium as a volume diffraction grating.

When a data page needs to be read out, the medium is illuminated with the reference beam at the angle that was used to record the desired page (Figure 3.a). The light is diffracted by the Bragg-matched grating within the medium into the original Fourier-transformed image recorded

with the same reference beam. The image detection system then inverse Fourier transforms the output which is received by a CCD camera and converted to its original form.

The process of reconstructing one beam by illuminating the medium with the other can be viewed as a symmetrical one. Consequently, illuminating the medium with a previously stored image will cause its unique reference beam to be reconstructed. Fast associative retrieval in a holographic memory system is, thus, possible because any reference beam can be reconstructed in the same manner that any image beam can. In a content–based search, a portion of a previously recorded page, the search argument, can be used to illuminate the medium, resulting in partial reconstruction of the reference beams of those pages containing the search argument (see Figure 3.b). These reconstructed addresses are then collected by the reference beam detection system and used by the electronic host to retrieve only the desired information by providing the appropriate angles for retrieving the qualifying pages.

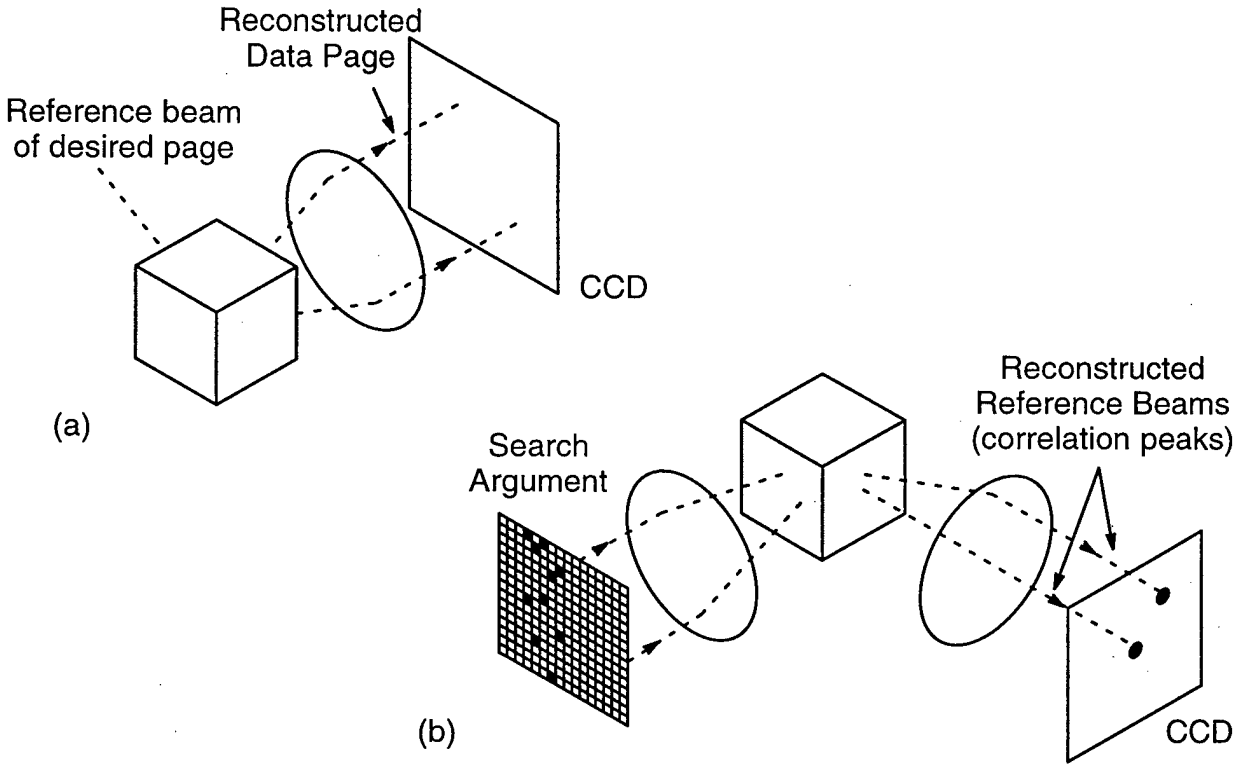


Figure 3: (a) Addressed-based retrieval and (b) associative search in VHDS.

Unlike many holographic associative memory implementations, ours is a two-iteration process: first the search, then the retrieval. Commonly, holographic associative memories use a non-linear threshold operation and a phase conjugating mirror to reflect the reconstructed reference beam back to the crystal [OWE89]. Although this allows reconstruction of the output at the same time the memory is being searched, it permits retrieval of only a single hologram: the one whose reference beam was reconstructed at the highest intensity. Our system allows multiple matches to be found, as would be necessary in a database environment, in a single step by thresholding the output first, potentially detecting several peaks, and then reconstructing the pages sequentially.

3. Data Encoding

While many different methods for encoding digital data exist [HAM86, IMA90, PLE89], page-oriented memories, such as holographic and other optical memories, have unique characteristics which require new coding schemes for implementing effective error detection and correction [DEP97]. First, the code must utilize the 2-dimensional bit arrangement and large page size of holographic memories. Current error-correcting codes (ECCs) are often applied to relatively small, linear groups of bits and their application to page sizes up to 10^6 bits is impractical. Second, the code must allow for fast error detection and correction in order to not significantly diminish the speed gained from the parallelism of the memory. Many ECCs, however, perform the final check in a bit-serial manner.

In our case, the code had to be applied to a specific type of data. In general, relational data are arranged in a tabular format and consist of a restricted character set. These characteristics can be incorporated into the encoding scheme to improve and simplify the code.

Error-control codes perform their function through the addition of bits, called *redundancy bits*, to the original binary data. The redundancy bits are created in a systematic way such that when one or more errors occur, the received data can be recognized as not being a proper codeword and the errors can be detected and/or corrected, depending on the type of code used.

Three parameters are used to describe the number of data and redundancy bits in the codeword: the total number of bits per word, n, the number of data bits per word, k, and the *rate*, $r = k/n$. The code rate is often used as a figure of merit for the code, indicating how much information is contained in each codeword. A code is often described by the triplet (n, k, d) with rate r, where d is the minimum Hamming distance between any two words in the code.

3.1 Array codes and the cross-parity code

One of the oldest and definitely the simplest ECC is the familiar single parity check. This code is, of course, of little practical use since it only detects a single error and provides no method of data correction. In 1950, R. W. Hamming developed a single-error-correcting–double-error-detecting (SEC–DED) code based on the systematic specification of parity check groups within the data word. When these parity checks are made on the received data, a group of bits called a *syndrome* is generated. The syndrome points to the exact location of the bit in error, provided only one error has occurred. Use of an overall parity check can indicate the occurrence of a double error. Hamming codes have since provided the basis for many other more complex and powerful codes.

In 1961, Hamming's original concept of parity check groups was applied to the rows and columns of a rectangular array of bits in an effort to provide simple decoding and low redundancy. This 2-D parity checking concept has since led to the development of a class of codes called *array codes*, which have found application in many different areas of both data communication and storage [FAR79, FAR82, FAR90, FAR92, MAB91]. The development of array codes arises from the geometric combination of other codes, such as the combination of row and column parity checks. In this way, researchers have been successful in creating effective and relatively simple codes for many applications.

Most of the more sophisticated array codes perform the final check and correction in a bit-serial manner and thus do not satisfy our criterion of parallel decoding. However the original row/column parity check code, called the cross-parity code (CPC), has the capability of being

decoded in parallel by an array of logic gates. This means that the decoding process can take place in O(1) clock cycles, meeting our fast decoding goal.

Figure 4 shows an example of the CPC. Each bit in a rectangular array of I rows and J columns is denoted by $b_{i,j}$, where i and j are the row and column indices ranging from 1 to I and 1 to J, respectively. When writing data, two vectors of parity bits are generated; an I-bit vector with each bit, p_{ri}, representing the parity bit for row i, and a J-bit vector with each bit, p_{cj}, representing the parity bit for column j. These vectors are appended to the data bit array along with an overall parity bit, p_{all}. An $(I+1) \times (J+1)$ array, called a *codeblock*, is thus formed. In the case of a page-oriented memory, many codeblocks are then grouped together, forming a data page to be stored, as illustrated in Figure 5.

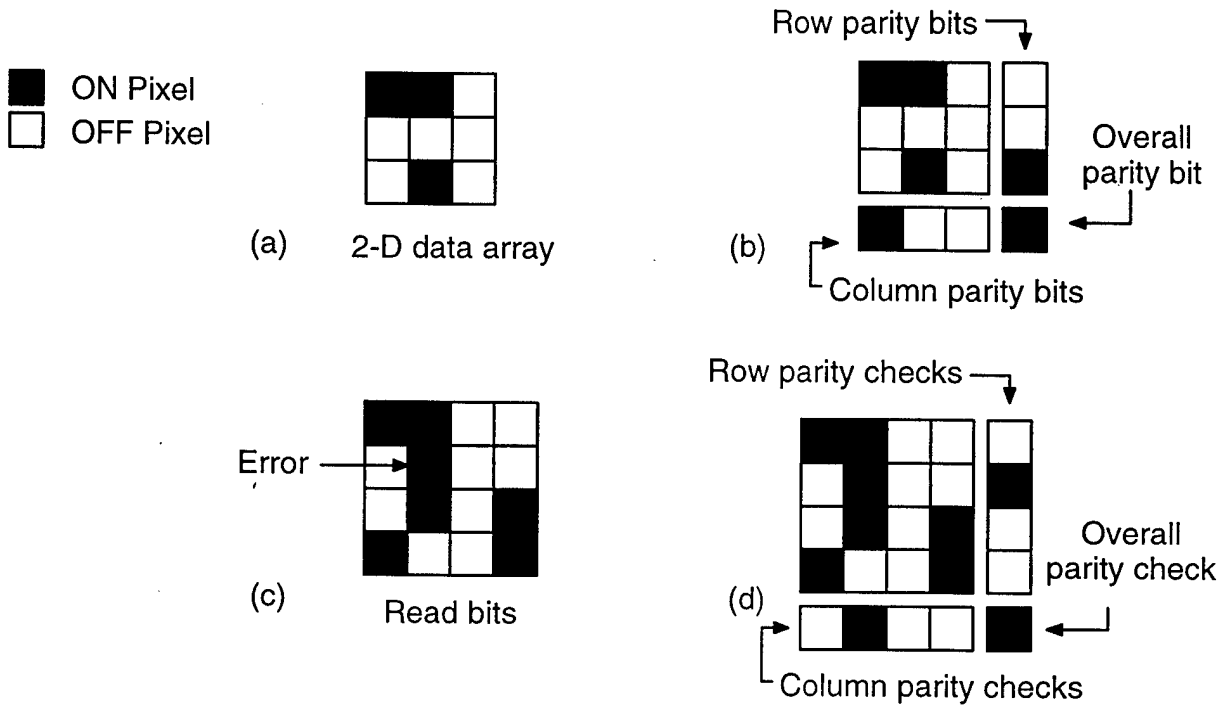


Figure 4: Example of the cross-parity code for a 3×3 array of data bits. (a) Data bits. (b) Data array with parity bits appended to be stored in memory. (c) Data and parity bits read out, error in center data pixel. (d) Intersection of ones in parity check vectors indicates the error.

When a data page is read, parity checks are made on each of the $I+1$ rows and $J+1$ columns of each array. The occurrence of a single error in bit $b_{i,j}$ will be indicated by a 1 generated by the parity checks of both row i and column j. The parity check signals are denoted by $p_{check,ri}$ and $p_{check,cj}$ for the row and column checks, respectively.

The CPC is a distance-4 code and can thus be used in a SEC-DED mode. Single-error correction is performed as in the example while double-error detection is accomplished by using the overall parity check [FAR79]. When a single error occurs, $p_{check,all}$ becomes 1 and a single 1 will appear in each parity check vector. However, when a double error occurs, 1's will occur in the parity vectors, but $p_{check,all}$ will be 0. Therefore a double error signal (DE) can be generated according to Eq. (1). The summation operators indicate the logical OR over the bits in a vector.

$$DE = \left(\sum_{i=1}^{I} p_{\text{check,ri}} + \sum_{j=1}^{J} p_{\text{check,cj}} \right) \cdot \overline{p}_{\text{check,all}} \tag{1}$$

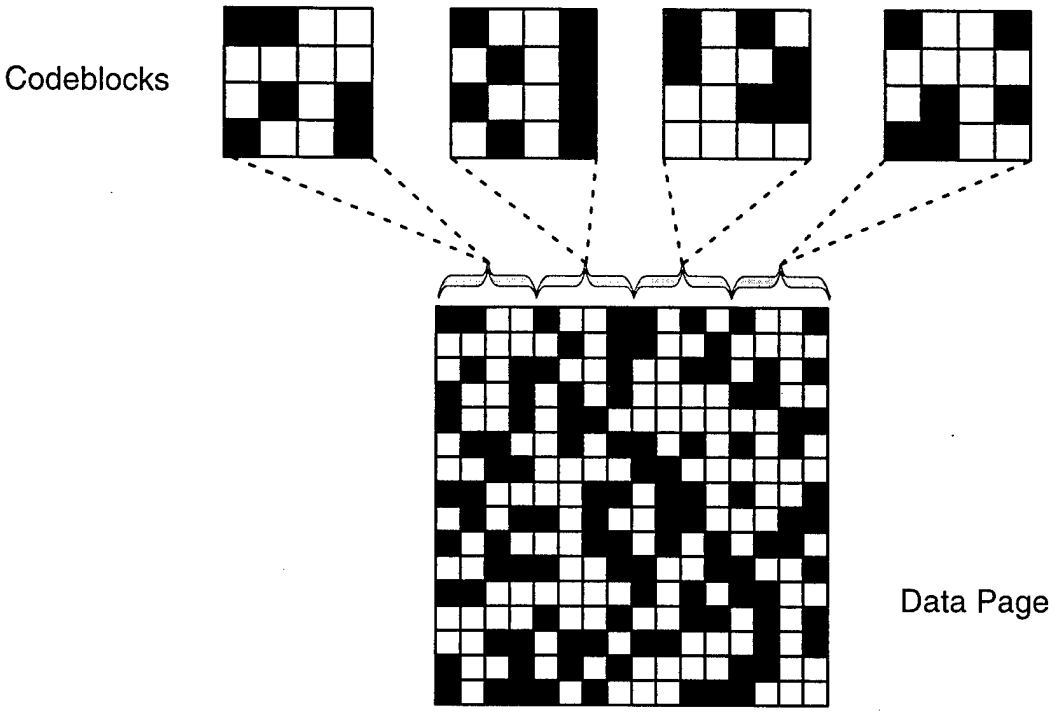


Figure 5: Example of four codeblocks combining to create the top four rows of a data page. Twelve more block are used to finish this particular 4×4 codeblock array.

The occurrence of *DE* indicates the presence of an uncorrectable double-error in one of the blocks on the page and, thus, the page must be reread. While the cross-parity code is only guaranteed to detect any two errors within an encoded block, the *DE* signal may also detect a larger even number of errors within the block, depending on the pattern in which the errors occur. Forcing a reread to occur upon double-error detection increases the power of the code, but also has a detrimental effect on the response time of the system. This issue will be discussed further in section 4.3.

A desired feature of any ECC is that it be self-checking. This means that an error occurring in the redundancy bits does not adversely affect the error correcting capabilities of the code. In the cross-parity code, a single error occurring in a parity bit vector can be corrected in the same manner as the data bits. However, in such a case there is no need to actually correct the parity bit, but only to detect it, since no data bits are in error. This fact will reduce the complexity of the error detection and correction circuitry.

3.2 Modified cross-parity code

Several modifications can be made to the general CPC to obtain more desirable properties. One of these is to require a constant number, *C*, of bits in a column to have the value 1. Although this restriction reduces the effective storage capacity due to a lower number of permissible bit patterns, it also exhibits several advantages. First, the need to store column parity bits is eliminated since they will always have the same known value. Second, selecting a small *C* relative to the total number of bits in each column results in a sparse encoding, yielding a higher

reconstruction contrast ratio [NEI93]. Finally, it helps in setting the threshold that can be used to determine matches during associative searching.

The cross-correlation of two images $a(x,y)$ and $b(x,y)$ is given by

$$U(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} a^*(x',y')b(x'+x,y'+y)\,dx'dy' . \qquad (2)$$

If we consider a and b to be two binary codewords with exactly the same number of ON pixels, it is easy to see that the correlation of any two codewords such that $a=b$ will have the same peak value, U_{eq}. Likewise, the correlation of any two codewords such that $a \neq b$ will also yield the same lower peak value, U_{ne}, where $U_{ne} < U_{eq}$. This greatly simplifies the determination of a threshold for distinguishing matches from non-matches. Although the peak intensity will still be dependent on the number of characters in the search argument, the character-dependency is removed.

Since the number of different characters in a database environment may be limited to about 50 (26 characters, 10 digits, and miscellaneous symbols and punctuation), further restrictions to the value of C can be made. The most obvious value would be C=1, resulting in a mapping of each ASCII character to 1 out of 50 row positions. However, this yields a rather large redundancy (50/8), lowering the system capacity. By choosing C=2, a mapping similar to that in Figure 6 can be created. One bit in the first five rows and one in the last ten rows of each data column is ON, creating 5 groups of 10 characters. The CPC is then applied to groups of J (J = 7 in this example) columns (characters). This scheme, termed CPC-VHDS, has a much better redundancy (15/8) [GOE95a, GOE95b].

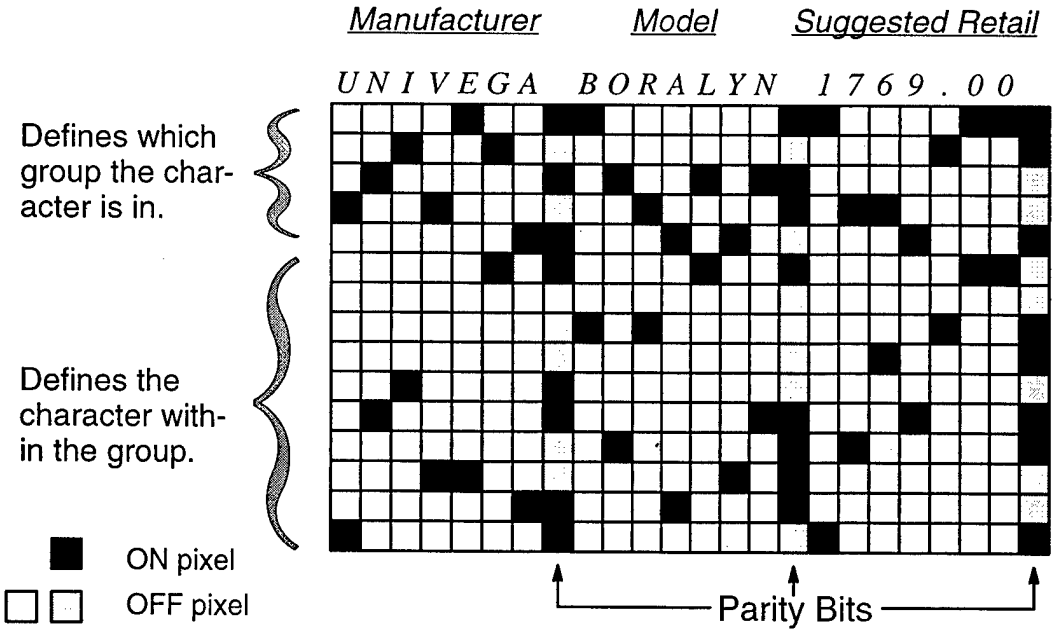


Figure 6: Cross parity code applied to blocks of 7 characters. No column parity bits are needed.

The selection of which bit pattern represents each character is somewhat arbitrary. However, it is preferable to avoid encoding methods which create similar patterns within and between pages. An example of an undesirable scheme is one that encodes all digits in the same group of 10, thus creating a horizontal line across every numeric field (such as phone numbers) in the

database. Such a common pattern may reduce the effectiveness of associative searching because each numeric field in the search argument would partly correlate with every numeric field in every record, resulting in more difficult detection of the proper correlation peaks at the output plane.

Although the subsequent discussions will generally focus on the C=2 encoding method shown in Figure 6, we have also investigated variations of CPC-VHDS which utilize larger values of C. Increasing C provides more freedom in the total number of allowable characters in the character set and allows more flexibility in developing simple, fast decoding hardware. Table 1 summarizes some of the characteristics, including code rate, of several different methods. Since each n-bit data word encodes one ASCII character, the code rate of CPC-VHDS is

$$r = \frac{8}{n}\left(\frac{J}{1+J}\right) \tag{3}$$

where n is the total number of bits per character.

C	n	Placement of 1's	Total Characters	Rate (J = 7)
1	50	One 1 per character	50	0.14
2	15	One 1 in top 5 bits, One 1 in bottom 10 bits	$5 \times 10 = 50$	0.47
3	12	One 1 per 4 bits	$4^3 = 64$	0.58
4	12	One 1 per 3 bits	$3^4 = 81$	0.58

Table 1. Comparison of several variations of CPC-VHDS.

The code rate numbers in Table 1 are comparable to encoding methods used in current optical memories such as CD-ROMs, which have effective code rates of 0.54, and bi-phase encoding methods like the Phillips 4/15 code, which has a rate of 0.53 without considering additional ECC bits [MAR90]. The restrictions placed on the character set allow a sufficient code rate to be maintained while utilizing a sparse encoding to achieve a higher SNR. The loss of generality resulting from these restrictions is acceptable since the system is intended for use in a database environment only.

Other codes for 2-dimensional optical memories have been proposed [NEI94a, DEP97] which provide some degree of decoding parallelism and either better code rates or greater error correcting capability. One such method, however, involves decoding a 1-dimensional word in sequential steps with 2-dimensional decoding performed with an array of 1-D decoders [NEI94a]. Another system employs bi-phase encoding (2 physical bits used to record each logical bit) of bit-slice pages in order to reduce the effect of burst errors [HEA94]. This approach requires multiple pages to be read and buffered before an entire word can be decoded. In our encoding scheme, decoding occurs completely in parallel, correcting all correctable errors on a 2-D page in a single step. These code properties are summarized in Table 2.

Code Author	Code Rate	Parallel Decoding	Data Restrictions	Error Correction Capability
Heanue [HEA94]	0.33	None	None	Correct 1 error per 12 bit block
Neifeld [NEI94a]	0.60	Word-Serial	None	Correct up to 12 errors per 60 bit block
Goertzen [GOE95a]	0.47–0.58	Fully Parallel	Alphanumeric and symbols	SEC-DED per 120 bit block

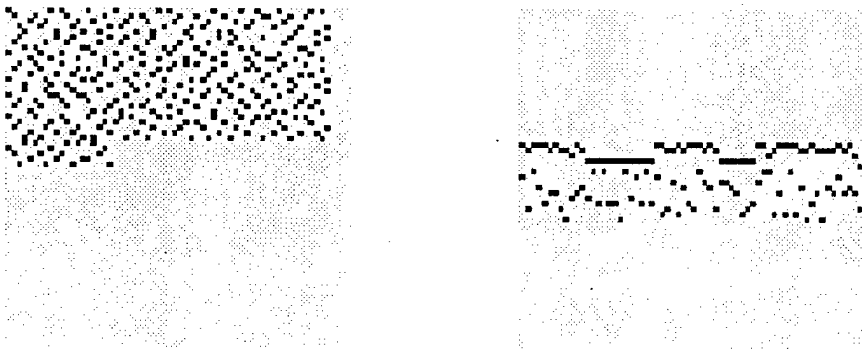
Table 2. Comparison of several proposed error correction codes for holographic memories.

Methods such as data interleaving can be incorporated into CPC-VHDS to account for burst errors, but at the expense of decoding simplicity and speed. Additionally, CPC-VHDS provides advantages in associative retrieval and increased reconstruction contrast ratio, as discussed earlier, and is easily scalable to different page sizes in different systems. These advantages make this code particularly suitable for application to database storage.

3.3 Output simulations

Simulations of the optical output were conducted with software which we developed [GOE96a]. Their purpose was to provide a visual indication of what the data pages may look like and, more importantly, to provide some indication of how the encoding scheme would affect the associative search process.

Figure 7 shows two data pages encoded with the same information using two different types of encoded characters. Figure 7(a) depicts a page with each character encoded as a 5×5 bit block where exactly one bit is ON in each row and column of the block. Characters fill the page by rows starting at the top left. The pattern in Figure 7(b) contains the same data encoded using the 2-of-15 characters described for CPC-VHDS above (though no parity bits are included here). Characters fill the middle rows of the page in this case.



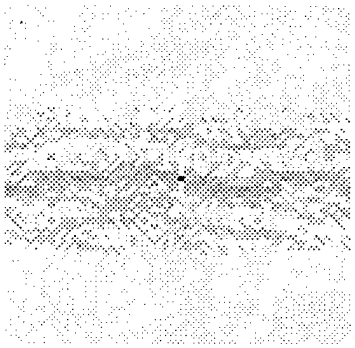
(a) 5×5 characters (b) 2-out-of-15 characters

Figure 7: Two pages containing the same information but with different encoding methods.

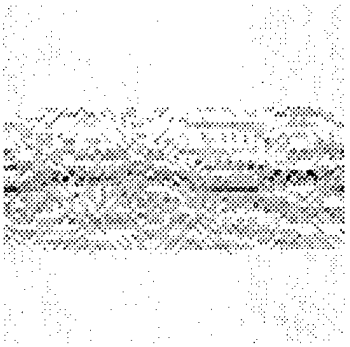
To observe how the encoding scheme will affect the associative searching process, the correlation between the pages in Figure 7 and several search arguments was calculated. The results

of these simulations are shown in Figure 8. The *Max. intensity* values given in the figure captions are intensities of the brightest pixel on the page in arbitrary units. The dark spots in the center of images (a) and (c) indicate high correlation, while the absence of this spot in (b) and (d) indicates low correlation. The appearance of dark areas on the pages with low intensity (b and d) is due to the normalization of intensity values for display purposes.

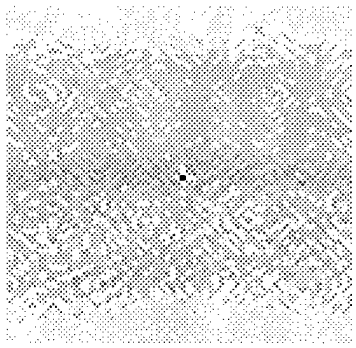
For the case of searching with a matching search argument (Figures 8(a) and (c)), we see that the peak intensity of the output using the 5×5 characters is about 2.5 times greater than for the page with the 2-of-15 characters. This is exactly what we would expect since each 5×5 character has 2.5 times as many ON pixels. In both cases the correlation peak is clearly detectable.



(a) 2-out-of-15 characters, correlation with matching search argument. Max. intensity = 3.9



(b) 2-out-of-15 characters, correlation with non-matching search argument. Max. intensity = 1.9



(c) 5×5 characters, correlation with matching search argument. Max. intensity = 9.8

(d) 5×5 characters, correlation with non-matching search argument. Max. intensity = 3.6

Figure 8: Correlation patterns for matching and non-matching search arguments under different encoding schemes. Darker areas indicate higher intensity.

Figures 8(b) and (d) show the results of the search using a non-matching search argument. These allow us to find the ratio of matching to non-matching peak correlation intensities. We see that this ratio is about 2.7 for the 5×5 characters and around 2 for the 2-of-15 characters. Since a larger ratio improves the system's ability to differentiate between matches and non-matches, it seems that investigation into coding methods incorporating a larger number of ones per character, such as the 3-of-12 and the 4-of-12 codes listed earlier, may prove beneficial.

3.4 Decoding hardware

Decoding each cross-parity encoded block can be performed in parallel with an array of XOR gate strings or trees, one for each row and column of each block on the page. In this manner, the parity checks for all blocks can take place concurrently. The simple and regular structure of the decoding hardware easily translates to a smart pixel array implementation [IRA95]. Figure 9(a) shows a cascaded gate structure for parity checking. In a smart pixel implementation, each pixel performs two XOR operations with a given input bit; one with the output of the previous row and one with the output of the previous column, as depicted in Figure 9(b). The final parity checks are then the outputs of the last pixel in each row and column.

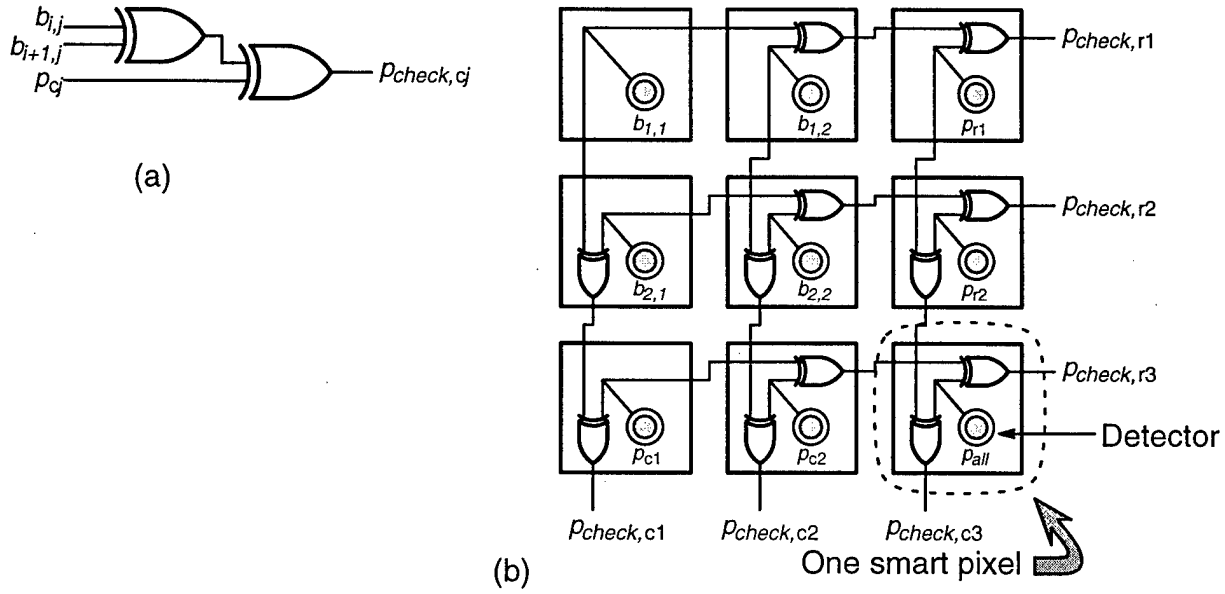


Figure 9: (a) Cascaded gate structure for parity checking. (b) Smart pixel array implementation.

An alternative approach involves a gate tree structure as shown in Figure 10(a). In this arrangement, each pixel receives four inputs and performs four XOR operations; two vertically and two horizontally, as in Figure 10(b). The operations which take place within each pixel constitute only the first stage of the gate tree. The outputs of the pixels pass through the rest of the gate tree electronically to form all parity checks. In both cases, the arrays are easily expanded by simply adding more pixels to the array.

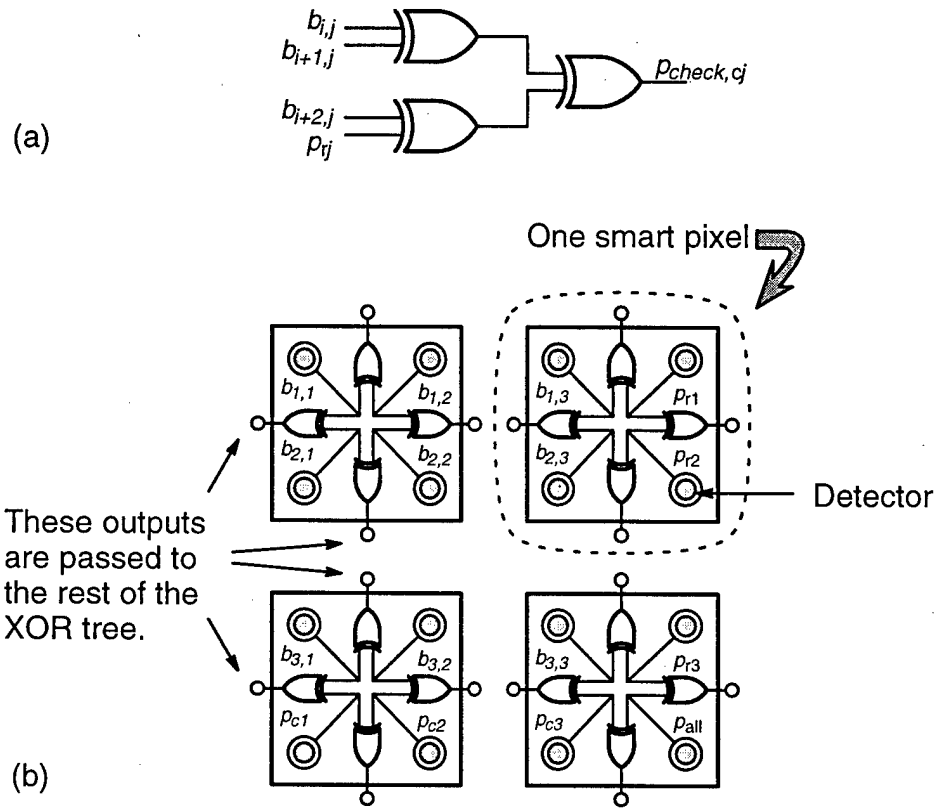


Figure 10: (a) Gate tree structure for parity checking. (b) Smart pixel array implementation. The XOR gates on each pixel are the first gates in the tree. The outputs of these gates are passed on to the rest of the XOR tree electronically.

The data correction hardware consists of two gates per bit, as shown in Figure 11. When a row and column parity check fail, the output of the AND gate will be one, indicating an error in that bit. In such a case, the result of the XOR operation will be the inverse of the erroneous data bit. The *DE* signal has no function in this circuit because when it is active, the data will be discarded anyway.

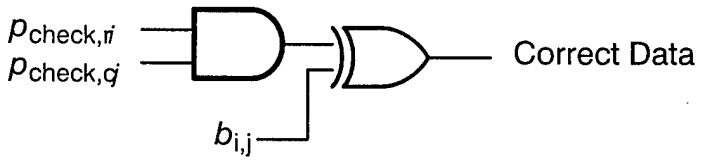


Figure 11: Bit correction hardware.

3.5 Error-correcting capability

The final or corrected BER of the system, *CBER*, for a given ECC is determined by several factors: the raw BER of the system, *RBER*, the total number of bits per block, n, and the error detecting/correcting capability of the code (i.e., the number of errors per block which can be detected and/or corrected). Based on these, a model for *CBER* was developed so that a maximum acceptable *RBER* for the system could be determined. This model assumes a random noise source introducing random errors. System issues, such as lens aberrations, SLM pixel defects, or dust, may cause statistically dependent (non-random) errors to occur. The data interleaving methods mentioned earlier can be utilized to minimize these effects if they cannot be solved otherwise.

For a block of size n, the probability of no errors occurring is given by

$$P_0 = (1 - RBER)^n . \tag{4}$$

Expanding on this, the probability of zero errors getting through the decoding process, $P_{correct}$, can be written. Since the CPC is a SEC-DED code, this can be written as the sum of the probabilities of the occurrence of 0, 1, and 2 errors.

$$P_{correct} = (1-RBER)^n + nRBER(1-RBER)^{n-1} + \frac{n(n-1)}{2} RBER^2(1-RBER)^{n-2} \tag{5}$$

Since $P_{correct}$ is the probability that no errors get by the decoder, it can alternatively be written in terms of *CBER* in the same way that P_0 was defined.

$$P_{correct} = (1-CBER)^n \tag{6}$$

For any ECC, the error-correcting capability depends in part on block size. That is, for a fixed error-correcting capability and *RBER*, *CBER* will decrease for larger codeblocks. As mentioned previously, the code rate of CPC-VHDS increases with larger codeblocks. Thus, we see a common and important tradeoff between *CBER* and code rate.

To examine this tradeoff, we would like to optimize the code rate by utilizing the largest codeblock which will yield an acceptable *CBER*. Industry standards call for $CBER \leq 10^{-12}$. Substituting this value into Eq. (6) and setting Eqs. (5) and (6) equal to each other permits solving for the value of n which satisfies our restriction on *CBER*. Each of these n values then corresponds to a certain number of CPC-VHDS–encoded characters. Figure 12 gives a plot of the maximum number of characters, J_{max}, which can be included in a codeblock and still satisfy our *CBER* requirement for a given *RBER*. The data show that an acceptable *CBER* is attained for J=7 (as in our examples) for a raw BER around 10^{-5}. This result is encouraging since these raw BER values have been attained in holographic memory system demonstrations by several different groups [BUR95b, SHE95].

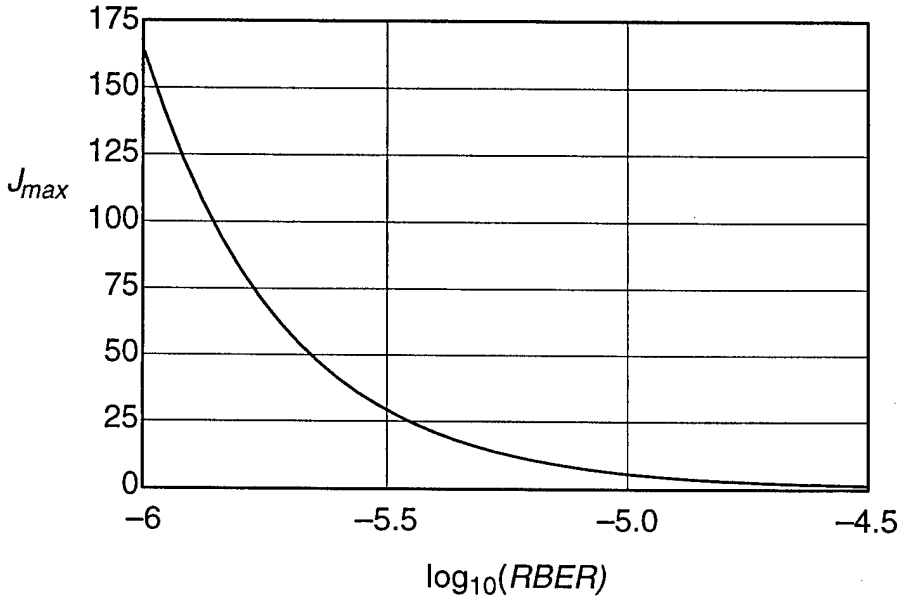


Figure 12: Maximum number of character per codeblock, J_{max}, which yields $CBER \leq 10^{-12}$ for a given *RBER*. The plot shows a rapid increase in block size, and thus code rate, as *RBER* drops below $10^{-5.5}$.

Similarly, we can solve for *CBER* as a function of *RBER* for a given block size. A plot of such a function is given in Figure 13. We see here the direct relationship between *RBER* and *CBER*, as would be expected. This plot also shows that the industry standard can be attained for a raw BER around 10^{-5}.

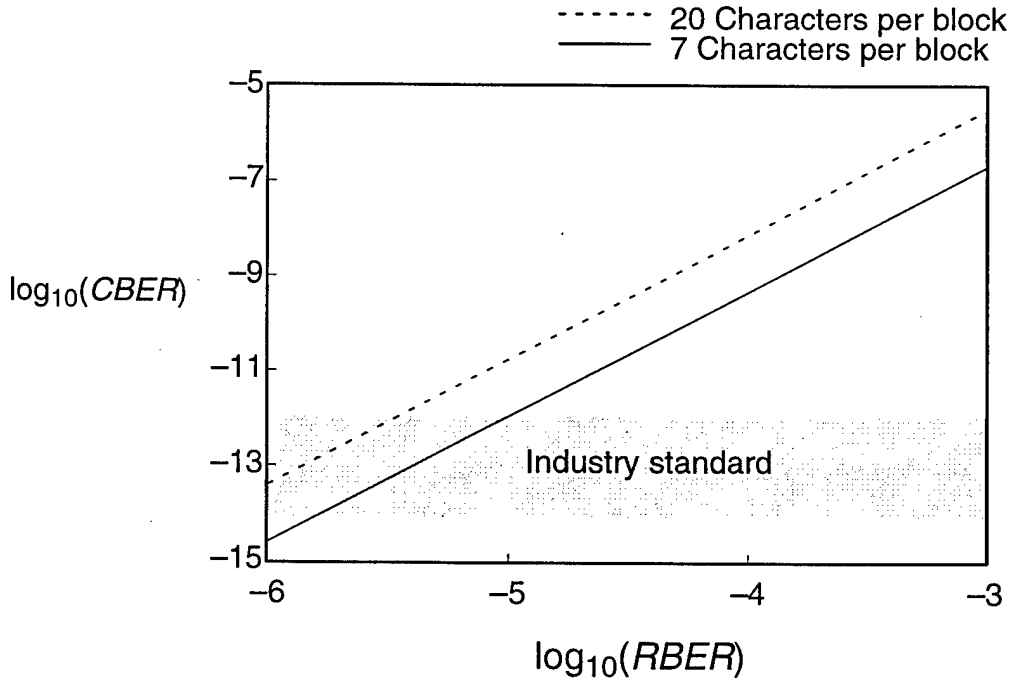


Figure 13: *CBER* as a function of *RBER* for different block sizes.

4. Performance Analysis of VHDS

We describe here the results of a theoretical performance analysis carried out on our design. The analysis focused on two areas: storage capacity and response time. These are the areas in which holographic memories promise to provide significant gains over other storage methods. The results of our analysis show that a volume holographic storage system may potentially store more than 10 million database records of 300 characters each. Also, the response time for associative retrieval can be more than an order of magnitude faster than current electronic machines.

4.1 Storage capacity

For a holographic memory system, the total storage capacity in bits can be represented by

$$N = N_{b/c} \times N_{c/p} \times N_{p/l} \times N_l , \tag{7}$$

where N is the total number of bits which can be stored in the system; $N_{b/c}$ is the number of bits in each character; $N_{c/p}$ is the number of characters on each page; $N_{p/l}$ is the total number of pages which are multiplexed in each location of the holographic medium; and N_l is the total number of spatial locations at which pages are stored.

N represents the total bit capacity of the memory. However, this will deviate from the usable storage capacity depending on how many bits are required to store each character. The effective storage capacity, N_{eff}, can be written as

$$N_{eff} = \frac{\#\ of\ bits\ per\ character\ in\ ASCII\ encoding}{\#\ of\ bits\ per\ character\ in\ ECC\ encoding} \times N \tag{8}$$

or

$$N_{eff} = 8 \times N_{c/p} \times N_{p/l} \times N_l \tag{9}$$

For CPC-VHDS with 7 characters per block and a page size of $D \times D$, Eq. 9 becomes

$$N_{eff} = 8 \times \left\lfloor \frac{D}{15} \right\rfloor \times \left\lfloor \frac{D}{8} \right\rfloor \times 7 \times N_{p/l} \times N_l \tag{10}$$

The plot of Eq. 10, given in Figure 14, shows that effective capacities approaching 5 Gbytes are possible for large page sizes if more than 50,000 pages can be stored (note that N_{eff} is plotted as Mbytes, not Mbits).

If the values in Figure 14 are considered reasonably attainable, the question is then how these capacities match up with the needs of database storage. Figure 15 shows a plot of the effective storage capacity needed to store database records of varying sizes. $N_{chars/rec}$ is the number of characters in each record and N_{rec} is the total number of records in the database. The data are plotted over the range of values where the storage capacity and the storage requirements overlap. The plot indicates that a holographic storage system as described above is potentially capable of accommodating a database of up to 10 million large records.

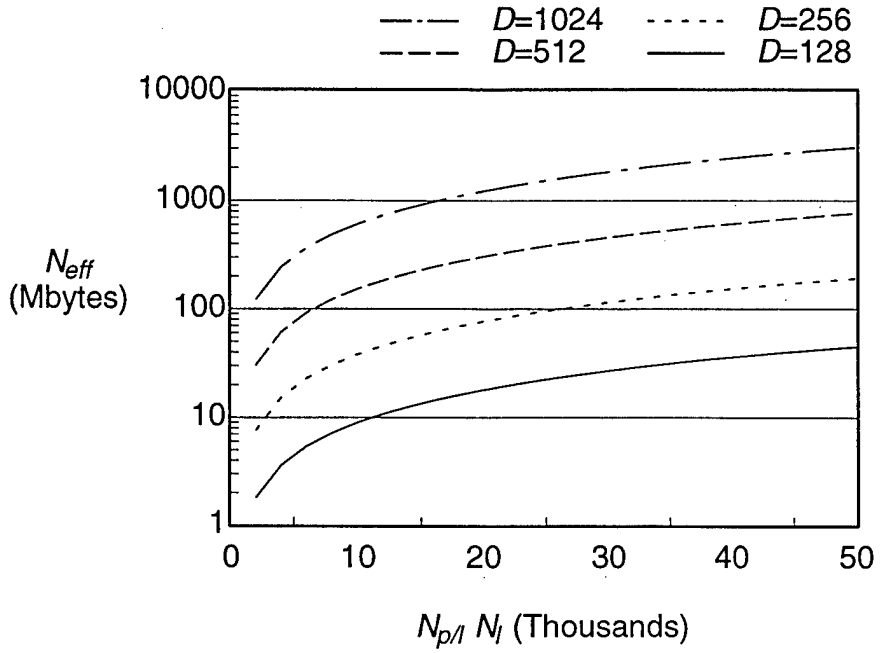


Figure 14: Effective storage capacity of a holographic memory for varying page sizes and number of stored pages.

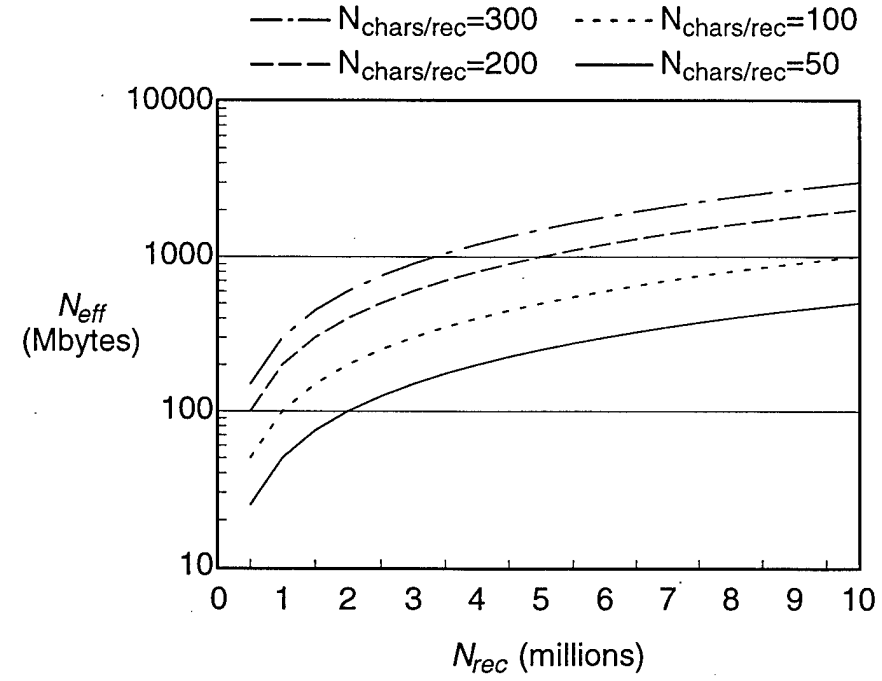


Figure 15: Effective storage capacity needed to store a given number of records of specific size.

4.2 Response time

Knowing that most operations in a database environment involve the retrieval of a record or group of records per request, it is more useful to discuss the response time of the system than the data rate, which is a commonly used performance metric. We define the response time here as the time between the moment a request for data is made and the time when the desired informa-

tion becomes available. This is directly affected by the system components, the type of data access (addressed or associative), and the possibility of having to reread a page of information due to double errors.

The response times of the system components are $\tau_{shutter}$, τ_{angle}, τ_{SLM}, τ_{CCD}, and τ_{decode} for the shutter, generation of the angularly-encoded reference beam, SLM, CCD detector array, and decoder, respectively. τ_{CCD} is the total response time of the CCD array which includes both the integration time (the time over which optical power is integrated on the array) and the time to read all pixels from the detector.

Address-based retrieval is performed by generating the reference beam (i.e. deflecting to the desired angle), illuminating the crystal, and then detecting and decoding the output. Thus the addressed retrieval response time, τ_{addr}, is

$$\tau_{addr} = \tau_{angle} + \tau_{shutter} + \tau_{CCD} + \tau_{decode} . \tag{11}$$

A fast deflector (such as an acoustooptic device) can be set in only a few microseconds and decoding can be done in a parallel fashion with less than a microsecond delay. The shutter, SLM, and CCD, however, have response times on the order of milliseconds. Thus τ_{angle} and τ_{decode} can be eliminated from Eq. (11) and the equation for τ_{addr} is approximated by

$$\tau_{addr} \approx \tau_{shutter} + \tau_{CCD} . \tag{12}$$

For associative retrieval, the search argument must first be generated on the SLM, the output reference beams must be detected, and finally each matching page retrieved by address. Thus τ_{assoc}, the associative retrieval response time, is the time to load the SLM with the search argument plus the time to detect the location(s) of the matching page(s) plus the time to retrieve and process those pages. Again ignoring τ_{angle} and τ_{decode}, τ_{assoc} is

$$\tau_{assoc} \approx 2\tau_{shutter} + \tau_{SLM} + \tau_{CCD} + \sigma N_{rec}(\tau_{CCD} + \tau_{post}) \tag{13}$$

where σ is the selectivity factor equal to the percentage of records which match the selection criterion ($\sigma \leq 1$), N_{rec} is the total number of records in the database, and τ_{post} is the time required to do any necessary post-retrieval processing to determine an exact match with the search argument.

This analysis is valid only for purely angularly multiplexed systems. If spatial multiplexing is also employed to increase capacity, the address based retrieval response time does not change, but the associative retrieval response time is directly affected. For spatio-angular multiplexed systems, the search process must be carried out for each of N_l locations. Thus, τ_{assoc} becomes

$$\tau_{assoc} \approx 2\tau_{shutter} + \tau_{SLM} + N_l\tau_{CCD} + \sigma N_{rec}(\tau_{CCD} + \tau_{post}) \tag{14}$$

where we have assumed that the response time of the deflector used to direct the search argument to the next location is on the order of τ_{angle}, and have neglected it in the third term.

Since the majority of retrievals in a database environment are content-based, we are primarily interested in τ_{assoc}. Figure 16(a) is a plot of τ_{assoc} versus τ_{SLM} for the angularly multiplexed case only. In this plot, σN_{rec} was taken to be one (i.e. only one matching record). Several values for τ_{CCD} (which varies with readout channel bandwidth and the number of parallel channels

[NOR94]) were used and $\tau_{shutter}$=3 ms was assumed. Since we assume only one record will be retrieved, the actual value of N_{rec} is not important. The search time in VHDS is independent of the number of records because the entire database is queried in a single step. The plot shows τ_{assoc} values ranging from 10 to 150 ms.

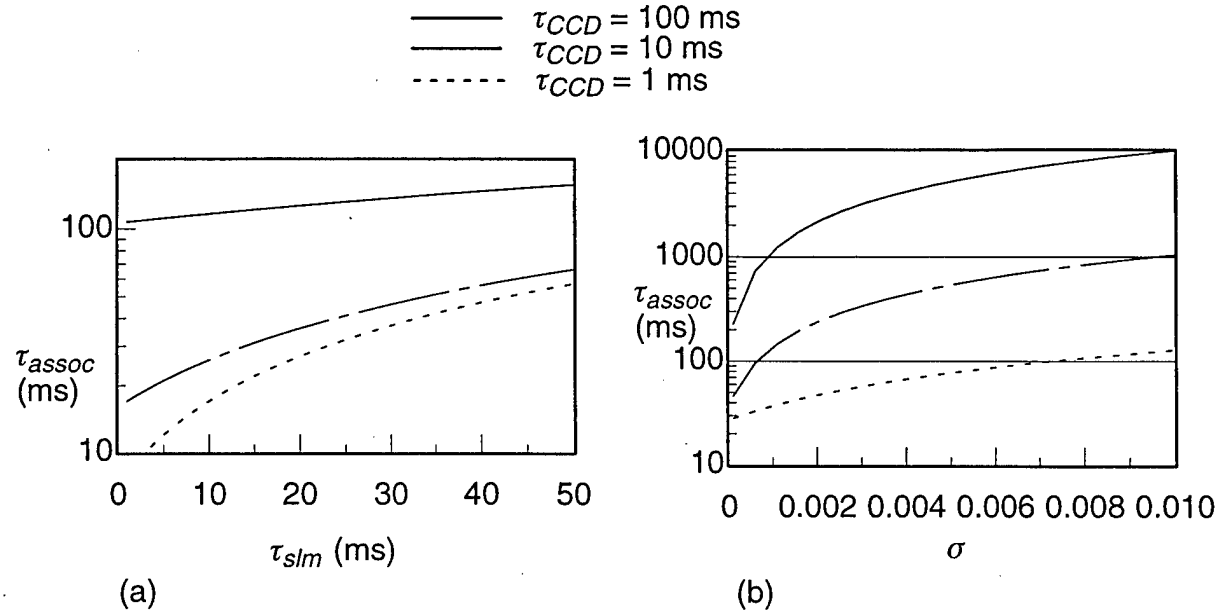


Figure 16: VHDS associative retrieval response time versus (a) τ_{SLM} for retrieval of a single record and (b) σ for retrieval of multiple records (τ_{SLM}=20 ms).

The case for retrieval of multiple records is shown in Figure 16(b). Here we have plotted τ_{assoc} versus σ assuming N_{rec}=10,000 and τ_{SLM}=20 ms. The plot shows that for a selectivity factor of 0.01 (retrieval of 100 pages), the response times range from about 0.1 to 10 seconds. Current electronic database machines exhibit response times for similar queries in the range from 12 to 120 seconds based on the type of search algorithm and number of search criteria used [ONE93]. It is important to note here that the search time in VHDS does not vary with the number of search criteria, unlike electronic database machines. That is, a search for the name 'Smith' and a search for both the name 'Smith' and the zip-code '68405' are performed equally fast since all records and attributes are searched simultaneously.

4.3 Rereads and average response time

The response time of the system is directly impacted by the need to reread pages after error detection. For VHDS, rereading is necessary when exactly two errors occur within one or more block on the page. The probability, P_2, of two errors occurring in a block of size n is given by

$$P_2 = \frac{n(n-1)}{2} RBER^2(1-RBER)^{n-2} \tag{15}$$

(again assuming statistically independent, or random, errors) where $RBER$ is the raw BER, as before. The probability, α, of at least one double-error occurring on a page is then

$$\alpha = 1-(1-P_2)^B \tag{16}$$

where B is the number of blocks on a page.

It is possible that a page which is reread due to double errors may need to be read again for the same reason. That page may then also need to be reread and so on. This effect is seen in the average response time of both addressed and associative retrieval. We will examine the effect on the average address-based response time, $\tau_{addr,ave}$, as an example here. The conclusions we draw from this analysis can similarly be applied to the associative retrieval response time since the fourth term in Eq. 13 is analogous to τ_{addr}. We can write $\tau_{addr,ave}$ as τ_{addr} plus the average amount of additional time required for rereads, as in Eq. 17. The probability of rereading a page exactly i times ($1 \le i \le \infty$) is given by the product of α^i, the probability of i consecutive double errors occurring, and $1-\alpha$, the probability that the i–th reread does not contain a double error.

$$\tau_{addr,ave} = \tau_{addr} + \sum_{i=1}^{\infty} \alpha^i (1-\alpha)(i\tau_{addr}) = \sum_{i=0}^{\infty} \alpha^i \tau_{addr} = \frac{1}{1-\alpha}\tau_{addr} \qquad (17)$$

Thus, the time for an address-based retrieval is increased by a factor of $1/(1-\alpha)$.

Figure 17 depicts how the factor $1/(1-\alpha)$ varies with *RBER* for two different page and block sizes. Naturally, as the page size ($D \times D$) and block size increase, the probability of a double error occurring also increases, thus raising the average response time. Figure 17 also determines the point at which the effect on the average response time is negligible, or

$$\frac{1}{1-\alpha} \approx 1 \,. \qquad (18)$$

The plot shows that this condition is met for $RBER < 10^{-5}$ for each block and page size, thus satisfying the restriction made earlier for attaining acceptable *CBER* values. That is, the same raw BER that provides a sufficient corrected BER will not significantly affect the average address-based or associative retrieval response time.

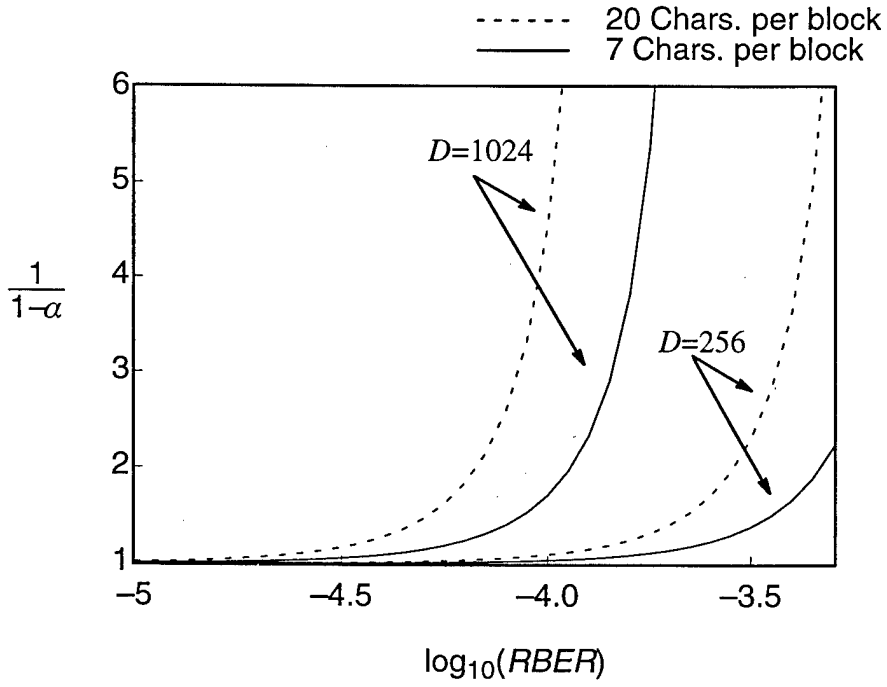


Figure 17: Effect of raw BER on the average response time. The response time is increased by a factor of $1/(1-\alpha)$. No significant effect is noticed for $RBER < 10^{-5}$.

The results of this performance analysis clearly show the tremendous potential of holographic storage technology. Large capacities are currently attainable in very small crystal volumes and those capacities will continue to climb with more advanced system components and architectures. Currently attainable response times for associative retrieval in such systems can be more than an order of magnitude faster than current electronic searches due to the ability to search the entire database (or very large portions in the spatio-angular multiplexed case) in a single step and the independence of the search time on the number of search arguments.

In our analysis of response time, we also studied the effect of rereads due to double errors on the average addressed retrieval response time. These results show no noticeable effect for systems with RBER $\leq 10^{-5}$. This not only provides us with an attainable goal, but matches our earlier restriction which provided acceptable CBER through error detection and correction.

5. System Architecture

5.1. Operational Overview

The type of memory that we have implemented is an angularly multiplexed volume holographic memory that uses a 1 cm^3 Fe:$LiNbO_3$ crystal as its storage medium and a 50 mW 532 nm Nd:YAG laser as its power source. The system can operate in three different modes: the *recording mode*, the *addressed recall mode*, and the *associative recall mode*. A block diagram of this system is depicted in Figure 18.

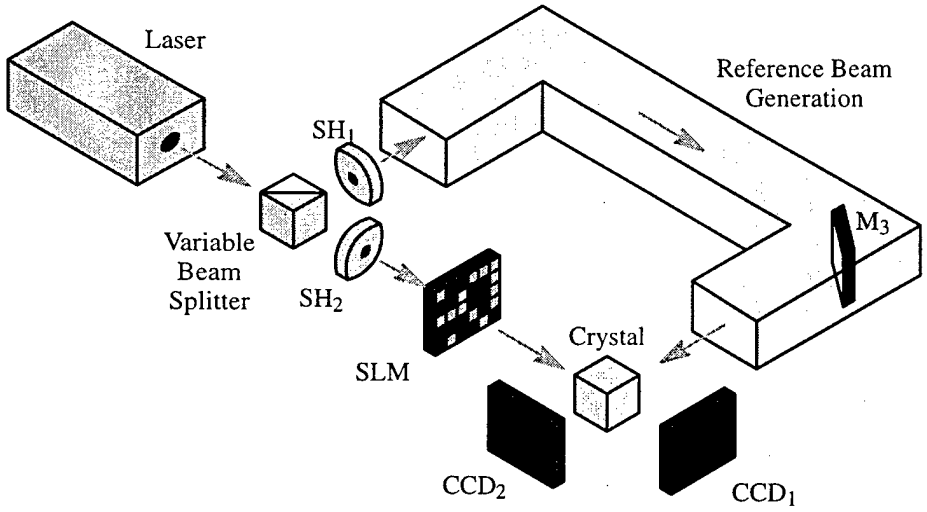


Figure 18. Functional diagram of VHDS with imaging lenses and collimating optics not shown.

5.1.1. Recording Mode

As the name indicates, in the *recording mode* data are stored in the memory. The process comprises the following steps: a) a pattern is loaded on the SLM from a computer, b) an address angle for the reference beam is generated by moving mirror M_3, and c) all three shutters (SH_1, SH_2, and SH_3) are opened for a predetermined amount of time, which is known as the exposure time. When the laser beam passes through the SLM, it is modulated by the information stored on the device and forms the contents of a page. This is the image beam and it enters one face of the storage crystal. The reference beam passes through the reference beam generation arm of the

system and enters another face of the crystal. The interference pattern between these two beams is stored within the crystal as a variation of its index of refraction. This process can be repeated many times thus storing multiple pages of data within the crystal as long as care is taken to prevent erasure of the contents of the crystal when new data are stored. When multiple holograms are stored in the same volume, each newly recorded hologram degrades the quality of previously stored data. We compensate for this degradation by recording the earlier holograms for a longer period of time, and then decrease the recording time for each subsequent hologram. This process is known as recording with an exposure schedule, and yields holograms that are all equally degraded and all at the same diffraction efficiency.

5.1.2. Addressed Recall Mode

The second mode of the memory, *addressed recall*, is used to retrieve data pages from the memory once they have been stored. This is done by illuminating the crystal with the reference beam at the angle that was used to record the desired data page. SH_1 and SH_3 are opened to allow only the reference beam to reach the crystal. A portion of this beam is diffracted by the modulated index of refraction within the crystal and gets directed towards CCD_1. The light that hits this camera is a reproduction of the original image beam used to store data at that unique address. Since the original image beam was modulated with a page of data, CCD_1 captures the desired data which can now be sent to an electronic host for further processing. This process can be repeated as needed to recall arbitrary data pages at random.

5.1.3. Associative Recall Mode

Finally, *associative recall* is the process that we use to search the entire contents of the memory in parallel. This is accomplished by loading the SLM with a search argument and opening SH_1 and SH_2. Due to Bragg selectivity, only pages that contain the search argument will reconstruct their reference beams with enough intensity to be captured by CCD_2. We call this holographic reconstruction of reference beam spots on CCD_2 the reference beam profile or the reference plane. Two examples of reference beam profiles are shown in Figure 19. The next step is to process the reference beam profile captured by the camera to find high intensity peaks that can be converted to hologram addresses. Visible in the two profiles of Figure 19 are bright spots that correspond to possible hits. The location of these peaks can be converted to actual addresses that can be used to retrieve the corresponding data pages with the system operating in the addressed recall mode. This process, too, can be repeated as needed.

To convert the location of the peaks to addresses, the first step is to actually obtain the location of the bright spots in terms of CCD pixel number. This is currently done by grabbing a single horizontal profile from the center of the reference plane and thresholding it to obtain values of either '0' or '1', where a '1' corresponds to a hit. We then convert the pixel location of each '1' into hologram locations. To aid in this step we record a unique image in at least three different hologram locations. If we search for this image we should get several hits each of which corresponds to known hologram locations. Using any two of these points will yield the equation of a line that can map pixel locations to page addresses. The remaining points can be used to check the validity of the mapping function. We can now use the mapping function to calculate the hologram address from each pixel location.

Figure 19. Two reference beam profiles obtained from VHDS-3.

The search argument can be any portion of a page from a maximum of 100% to a minimum percentage beyond which associative recall is not possible. Determination of this minimum percentage has been one of the objectives of this research.

5.2. Optical Overview

Now that we have an understanding of how an angularly multiplexed volume holographic memory works, let us examine the third and current prototype of our memory (VHDS-3) shown in Figure 20. This diagram shows every optical component in VHDS-3 and the following paragraphs will explain the purpose of each item. Subsequent sections will detail the differences between VHDS-3 and the earlier prototypes of VHDS-1 and VHDS-2.

The holographic memory is aligned such that the image and reference beams are incident to the crystal at a 90° angle. This geometry has been known to yield higher capacity and lower crosstalk [BUR95a].

The optical input to our system is provided by a vertically polarized, 50 mW, 532 nm, diode pumped, Nd:YAG laser. The laser beam passes through a safety shutter (SH_1) followed by a periscopic mirror (Periscope) that is used to adjust the height of the beam. Next the beam is split into two paths by a variable beam splitter (VBS) which is also used to control the amount of power that is directed into the reference arm. The output beams of the VBS are both vertically polarized.

Following the beam path through the reference arm, the next component is a shutter (SH_3) that is used to turn the reference beam ON or OFF. The natural divergence of the reference beam is slightly too large so an iris is used to obtain the desired diameter. Next the reference beam passes through a lens (L_1), a mirror (M_3), and another lens (L_2) before entering the 1 cm^3 Fe:$LiNbO_3$ photorefractive crystal (PRC). The iris, lens L_1, and mirror M_3 are mounted on a motor driven linear translation stage that travels in the direction indicated by the arrow. This setup, in combination with lens L_2, allows us to change the angle of the reference beam. Lens L_1 focuses the reference beam on M_3 which is positioned on the back focal plane of lens L_2. The front focal plane of lens L_2 goes through the center of the crystal. This arrangement ensures that the reference beam passes through the PRC in the same place no matter where M_3 is located. Moreover, as M_3 moves, the spot that lens L_2 images into the crystal also moves, which means that the angle at which the reference beam enters the PRC changes, thus permitting angular multiplexing. Currently we can only move M_3 a total of 25 mm and we choose to move it 30 μm between each recording, which means that we can store a total of 833 pages. After the crystal, the reference beam is imaged onto a camera (CCD_2) by the combination of three lenses (L_8, L_9, and L_{10}) and a mirror (M_4). The three lenses provide the correct demagnification to image the entire motion of M_3 onto CCD_2, and M_4 is used to overcome space limitations on our optical table. In a more optimum system M_4 can be removed and L_8, L_9, and L_{10} can be collapsed to a single lens.

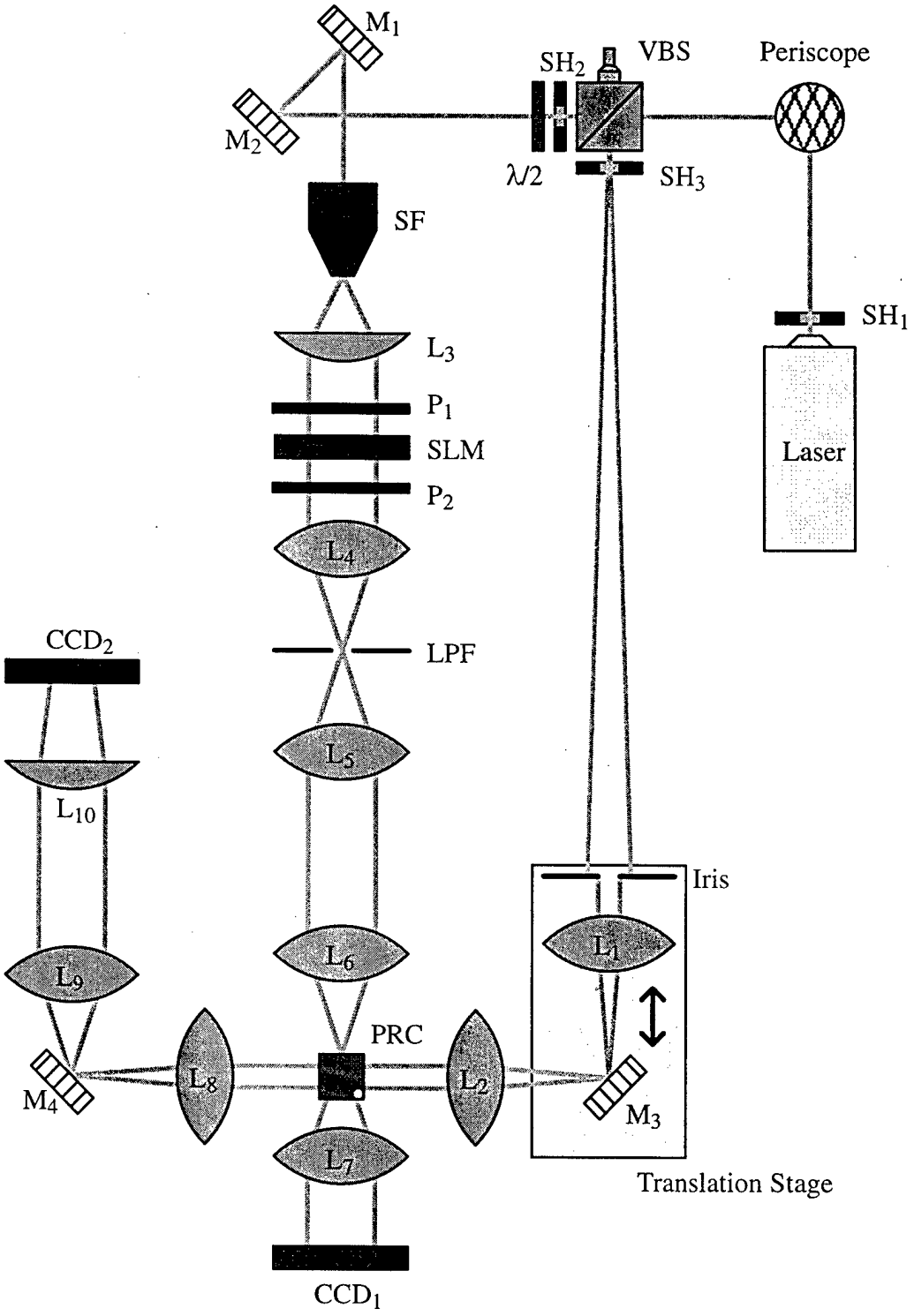


Figure 20. Optical components in VHDS-3.

The second output of VBS generates the image beam which passes first through SH_2 and a half waveplate ($\lambda/2$). SH_2 is used to turn the image beam ON and OFF and $\lambda/2$ can be rotated to give any linear polarization from completely horizontal to completely vertical, which, when combined with a polarizer (P_1), provides a method of adjusting the power that propagates through the image arm. Next the image beam passes through two mirrors (M_1 and M_2), that are used to adjust its trajectory, and a spatial filter (SF), whose purpose is to clean up and expand the beam. Lens L_3 collimates the beam and polarizer (P_1) allows only horizontally polarized light to reach the spatial light modulator (SLM). Polarizer (P_2) is adjusted to pass only vertically polarized light, a function that is necessary to achieve amplitude modulation of the image beam. The next three components, lens L_4, a low pass filter (LPF), and lens L_5, are used to filter out all but the zeroth order of diffraction that is generated when light passes through the SLM. Lens L_6 generates the 2D Fourier transform of the SLM pattern on the PRC. Finally, lens L_7 produces the 2D inverse Fourier transform of the light that exits the PRC and demagnifies it; this image is captured by the camera CCD_1.

The SLMs that we used were both ferroelectric liquid crystal displays made by Kopin with a resolution of 640 x 480 pixels. The first had a pixel size of 55 μm x 55 μm while the second was smaller with each pixel measuring only 24 μm x 24 μm. These transmissive mode SLMs modulate an optical beam by rotating the polarization of the incoming light. For example, if vertically polarized light passes through the SLM, then active pixels will rotate the polarization as much as 90° to produce horizontal polarization. Now if a polarizer is placed after the SLM that allows only vertically polarized light to pass through it, then just the light whose polarization was not rotated by active SLM pixels will pass through the polarizer. The SLM and polarizer combination permits the amplitude modulation of the light beam.

5.3. Automation

Early on in the course of this project, the need to automate most of the processes became apparent. Automation of the memory recording process was required to speedup the storage process and reduce the possibility of mistakes occurring during recording. To automate the system we need to be able to load the SLM, control the pulsing of the laser, and generate reference angles automatically. To accomplish these tasks, shutters were added to both outputs of the VBS as shown in Figure 18; a computer was dedicated to control the system; digital I/O circuitry was added to the computer; and software was either written or updated to open and close the shutters, to load the SLM, and to move the motor that repositions mirror M_3. These improvements have enabled automatic recordings, however, work is still underway to integrate CCD controlling programs and circuitry so that all three operational modes of the memory are fully automated.

Addressed recall mode is only partially automated. We currently have a program that controls the movement of the motor and the opening and closing of the shutters but cannot yet capture images from the CCD camera. Software has been developed to automatically extract data from captured images and generate intensity histograms and other useful statistics, such as SNR and BER measurements.

Automation of the *associative recall* requires three parts: a) automation of the searching process and, more specifically, that of capturing a reference beam profile, b) automatic detection of peaks in the reference plane, and c) automation of *addressed recall*, as described above.

6. Experimental Results

Through the course of this investigation, three prototypes of VHDS have been implemented by successively improving the hardware and control software. Table 3 summarizes the characteristics of each prototype.

Table 3. Prototype characteristics.

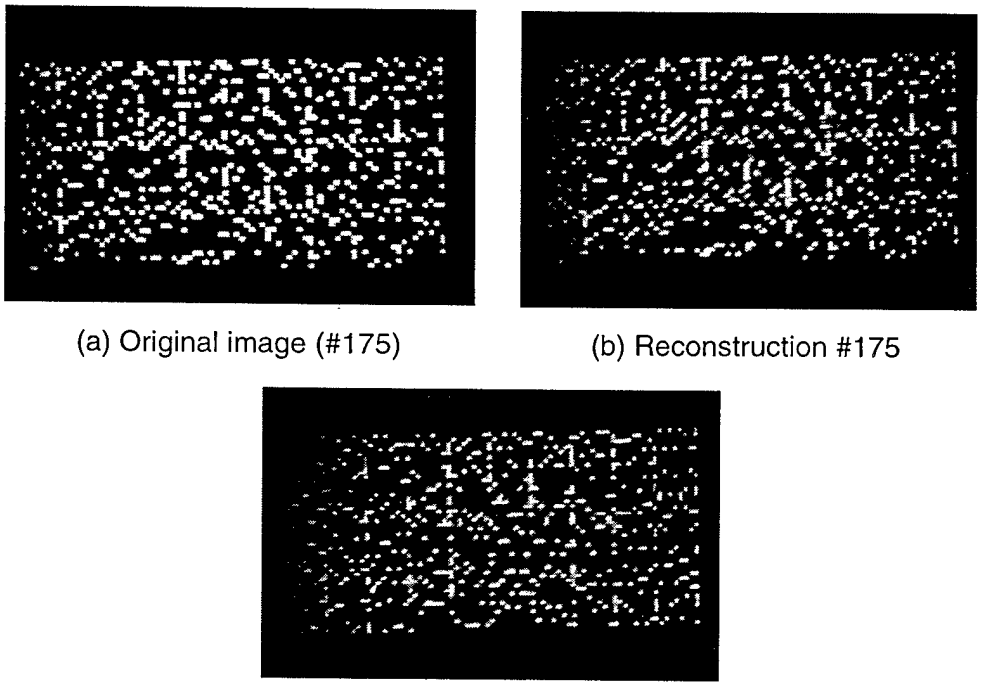
	VHDS-1 [GOE95b]	**VHDS-2** [RIC97a]	**VHDS-3** [RIC97a
Data Type	Random alphanumeric, Binary encoded	Real alphanumeric, Binary encoded	Real alphanumeric, Binary encoded Analog images, Grey–scale encoded
Maximum Pages per Recording	200	530	600
Maximum Characters per Page	210	98	210
SLM Pixel Size	55 μm x 55 μm	24 μm x 24 μm	55 μm x 55 μm
Pixels per Bit	5 x 5	11 x 11	Variable
Bit Size	275 μm x 275 μm	264 μm x 264 μm	Variable
Code Blocks per Page	3 x 10	2 x 7	Variable
System Automation	No Automation	Recording Automated	Recording Automated
Component Differences	Baseline	Added two shutters to perform automatic recording	Added two shutters to perform automatic recording. Added a half waveplate, a polarizer, and a variable beam splitter tuned to 532 nm to allow variable power levels in both the image and reference beams

6.1. VHDS–1 Results

The only process that was automated in the first prototype was the motion of mirror M_3 to generate addresses. Everything else (opening and closing the shutter, calculation of exposure times, blocking the undesired beam, loading the SLM, capturing images, and feeding back addresses) was controlled by the human operator.

In our experiments, 200 database records, one per page, were encoded using CPC-VHDS with 7 characters per code block (J=7) and recorded in the crystal. The angular separation be-

tween pages was 0.0023°. The ratio of peak reference beam intensity to peak image beam intensity was approximately 1, depending on the exact image stored. Each record (page) contained 210 pseudo-random alphanumeric characters arranged in fixed fields. The code blocks were placed on the page in three rows of ten blocks each for a total of 3600 bits per page, or a total of 720 Kbits. The total number of effective bytes (i.e. the number of characters) stored was 42 Kbytes. It should be noted that the page format described here was chosen to provide a reasonable page size. Figure 21 shows an example of one of the original images (21.a) and its reconstruction (21.b). The reconstruction of the first page (21.c) is also provided to show the extent of variations in the diffraction efficiency of the recorded pages.



(a) Original image (#175)

(b) Reconstruction #175

(c) Reconstruction #1

Figure 21: (a) Original image and (b) reconstruction of page #175. (c) Reconstruction of page #1.

We were also able to successfully reconstruct the reference beams of those pages which correlated well with an input search argument. Four of the recorded holograms (#11, 59, 116, and 162) were identical. Several different search arguments, ranging in size from 1/3 of the page to a single code block (1/30 of the page) were then used to search for these pages.

Figure 22 shows the results of the search using 1/3 of a data page. Figure 22.a shows the search argument and the output is given in Figure 22.b. Close inspection of the output reveals four intensity peaks, corresponding to the four matching pages. An intensity profile of the output is given in Figure 22.c. This profile verifies the correct location of the four matches. This plot also shows some diffraction efficiency drop in early holograms. This results in extra peaks in the thresholded output (Figure 22.d). Such a case is, however, acceptable when compared to the alternative of missing an output peak. It is better, in terms of reliable functionality, to have false hits in our search and retrieve several extra pages than to miss a matching page.

The results of the single codeblock search are shown in Figure 23. The search argument and outputs are given in Figures 23(a) and (b), respectively. The output appears to show four

intensity peaks, but the intensity profile and thresholded output reveal a fifth peak from the first hologram, which corresponds to a false hit. The experimental results obtained with VHDS-1 were published in references [GOE96a] and [GOE96b].

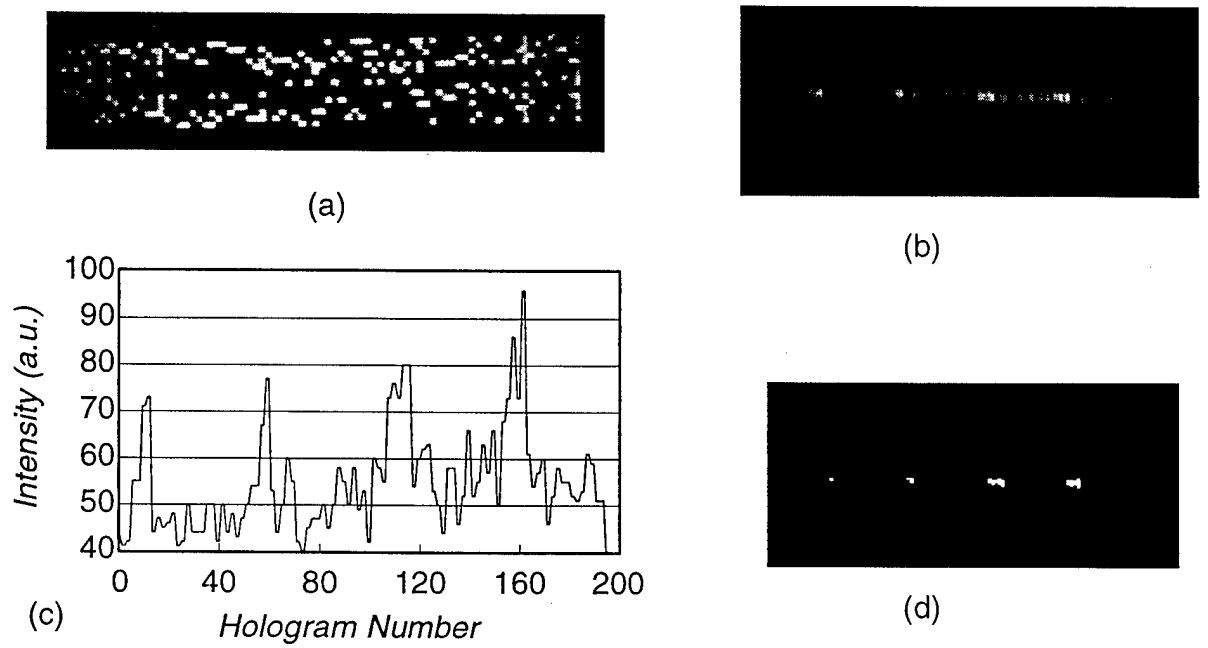


Figure 22: (a) Search argument, 1/3 of a data page, (b) Reconstructed reference plane, (c) Intensity profile along the center of the reconstruction, and (d) Thresholded reconstruction.

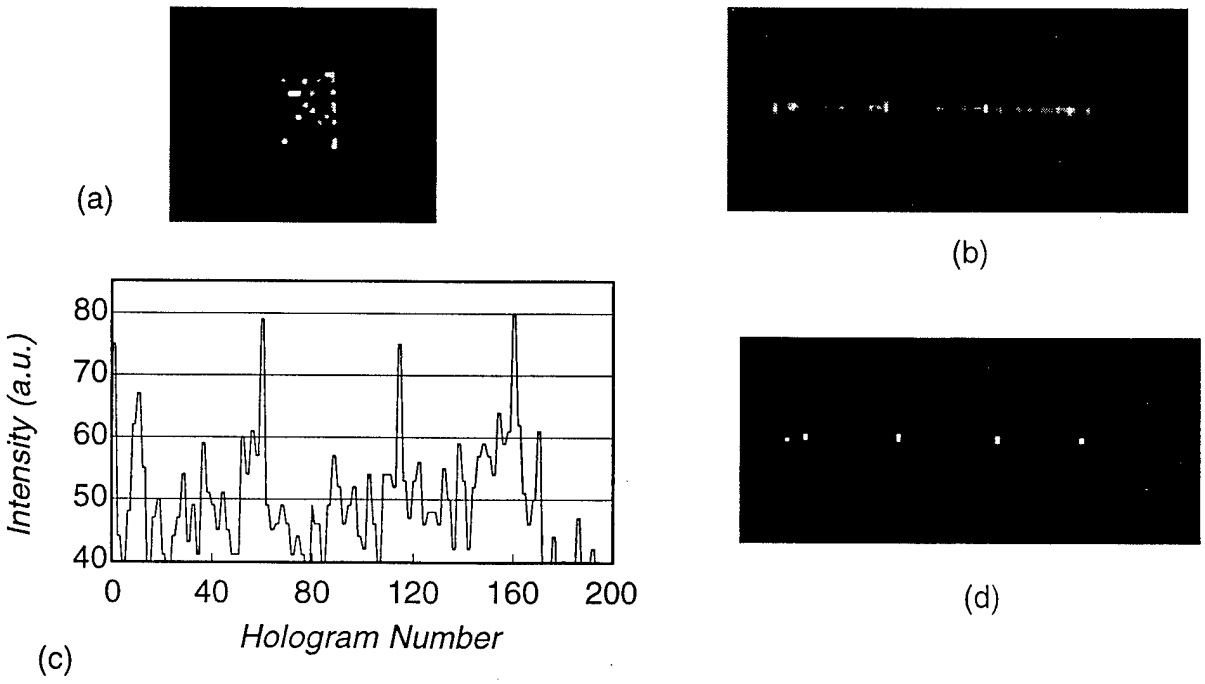


Figure 23: (a) Search argument, 1/30 of a data page, (b) Reconstructed reference plane, (c) Intensity profile along the center of the reconstruction, and (d) Thresholded reconstruction.

6.2. Second memory prototype

Automation of the recording process, use of a better quality SLM, and fine tuning of the optics have led to the second memory prototype (VHDS–2). We also increased the number of stored pages to 530 and used real data instead of pseudo-random data. Each page contained 98 characters comprising one record from a real database. The code blocks of each page were arranged in two rows of seven blocks for a total of 1680 bits per page [RIC96].

6.2.1. Database development

The use of random or even pseudo-random data does not allow an accurate evaluation of a memory system that performs matching based on data content. Modulated pseudo-random alphanumeric patterns produce data pages with a higher degree of orthogonality than what is expected with real data. This means that in order to better test the associative recall capabilities of our VHDS we should store real-world data. The data format that we chose comprised 530 records containing the last name, first name, company, address, city, zip code, telephone number, and e-mail address of SPIE/OSA members. With this database we now have an accurate representation of actual data that may be stored in a VHDS.

6.2.2. Diffraction efficiency experimentation

Many sets of experiments were performed to determine the exposure schedule for the second memory prototype. The exposure schedule that we desire is one which yields approximately equal diffraction efficiency for each reconstructed hologram during *addressed recall.* We have also observed that when the diffraction efficiency is not constant throughout a recording, the profile of the reconstructed reference beams does not have a uniform base value, and therefore, it is very difficult to detect the peak intensities that correspond to hits in associative searching.

6.2.3. Addressed recall results

To test *addressed recall* we illuminate the crystal with beams at angles that correspond to specific pages and then examine the memory output at CCD_1 to determine if the correct image is indeed recovered. The results from one such recall for the second prototype are given in Figure 24.

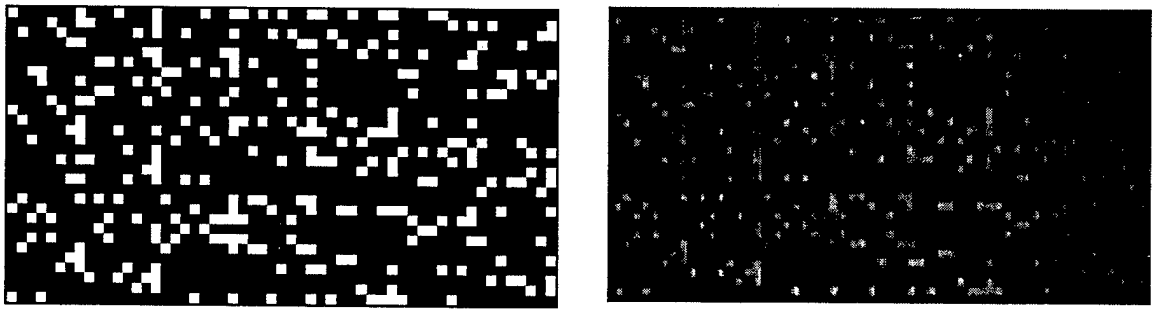


Figure 24. (a) Page stored at location 100, (b) image recalled from location 100 using VHDS-2.

6.2.4. Associative searching results

In testing the *associative recall* we explored how both the search argument and the data stored in the memory affect reconstruction of reference beam planes. Factors that were consid-

ered include the number of characters in the search argument; the number of matches; and the position, orientation, and size of the search argument. As can be seen in several of the plots below, during a read we may also generate some false hits. These do not pose a large problem though because in a relational database environment a small number of false hits can be expected and removed in post processing. Occasionally, we observe misses (pages that should have been selected and were not) that are more serious. Our efforts to improve the quality of reconstructed reference beam profiles aim towards reducing the number of false hits and eliminating misses.

As an example of the types of searches that we performed, consider searching through our database for the phone number of everybody whose first name is John. This is a rather small search argument (search argument size), it is very common (multiple hits), and John also appears in the last name of Johnson (position). These types of situations were presented to the memory to help us evaluate its performance.

To see how the number of characters in the search argument affects the reconstruction of the reference beams examine Figure 25. This figure shows the reconstructions captured by CCD_2 of a reference plane when searching for the hologram stored at address 323 using 98 and 15 characters. Notice in the profile plots that the intensity of the correct hit dropped from 0.84 to 0.25 when the number of characters in the search argument decreased. The main reason for this reduction in intensity is that, in the latter case, the total optical power illuminating the crystal is lower due to the smaller size of the search argument.

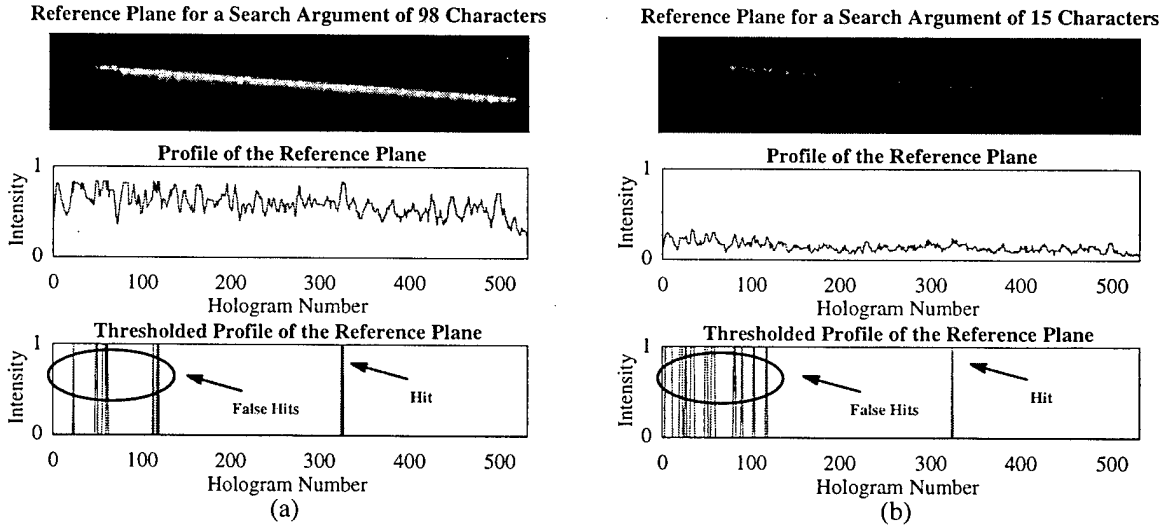


Figure 25. Reconstruction of the reference plane using (a) the entire record stored in location 323, and (b) only the last name in record 323 as the search argument.

We have also shown that it is possible to find multiple pages containing similar data. Figure 26 gives the results of a search for the first name (seven characters) that is contained within records 169, 269, and 394. These results show that we successfully found a match in page 169 and 394, but we failed to find the name in page 269. The profile plot below further demonstrates that as the search argument size decreases, the average intensity of the reference plane also decreases.

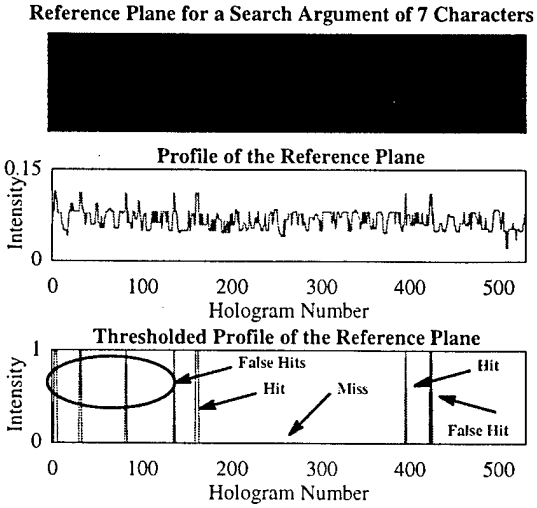


Figure 26. Results from a search for the first name that appears in pages 169, 269, and 394 using VHDS-2.

The last set of experiments performed shows that reference plane reconstruction is sensitive to the position, the orientation, and the scale of the search argument. Figure 27 provides examples when the search arguments were shifted, rotated, or enlarged versions of the last name in page 323. For all three situations the intensity was below the value of 0.24 observed in the profile plot of Figure 25(b). Also, a correct hit could not be indentified for any variations of position, orientation, or size that we tested. The conclusion here is that the search argument must appear in the original location, orientation, and scale in order for a match to be found. This behavior may be deemed undesirable in an associative memory used for target recognition but it is totally appropriate in a relational database environment where data selection is performed on well-defined and known a priori data fields.

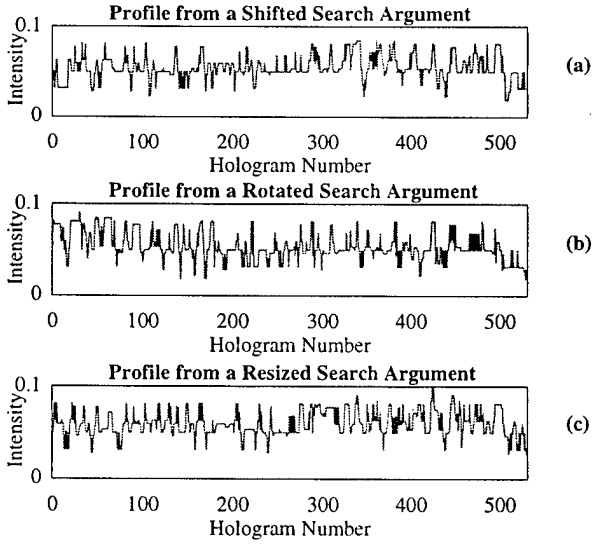


Figure 27. Results from a search for the last name that appears in page 323 when the search argument is (a) shifted, (b) rotated, and (c) resized.

The chart in Figure 28 summarizes some of the observations made using the second prototype. Here we can see that as the size of the search argument decreases, the average intensity of the reference plane also decreased. This means that detecting peaks becomes more difficult as the size of the search argument decreases. Also, notice that when no hit was found, there is still some intensity due to partial reconstructions as can be seen by the non-zero values in the last three bars. Finally, the average intensity when using only seven characters is almost as low as when no hit was found, meaning that for this system the minimum search argument size for successful associative recall is probably larger than seven characters. The above results were reported in references [GOE96a, RIC96, RIC97a, RIC97b].

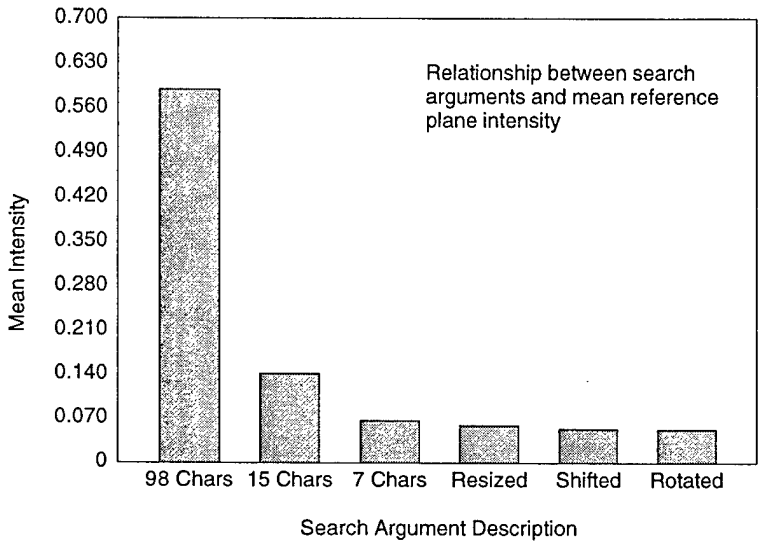


Figure 28. Mean intensity of reference plane profiles.

6.3. Third memory prototype (VHDS–3)

6.3.1. System improvements and *addressed recall* results

The second prototype seemed to have somewhat poor reconstruction of both the data and the reference plane so we decided to further improve the recording process for image clarity. VHDS-3 has the same basic structure as VHDS-2, however, we have upgraded some of the equipment. We have replaced the variable beam splitter with one optimized for the wavelength of our system (532nm) and we added a half-waveplate after the VBS in the imaging arm followed by high quality polarizers both before and after the SLM to compensate for polarization shifts introduced by the new VBS. This along with the realignment of optics has produced *addressed recalls* as shown in Figure 29. Figure 29.c plots the signal to noise ratio for the reconstructed pages in this experiment that involved recording 600 pages of real binary data. Notice that the quality of the reconstructions from both the second and third prototypes are about the same, however, VHDS-3 now stores 210 characters (3600 bits) per page whereas VHDS-2 only stored 98 characters (1680 bits) per page.

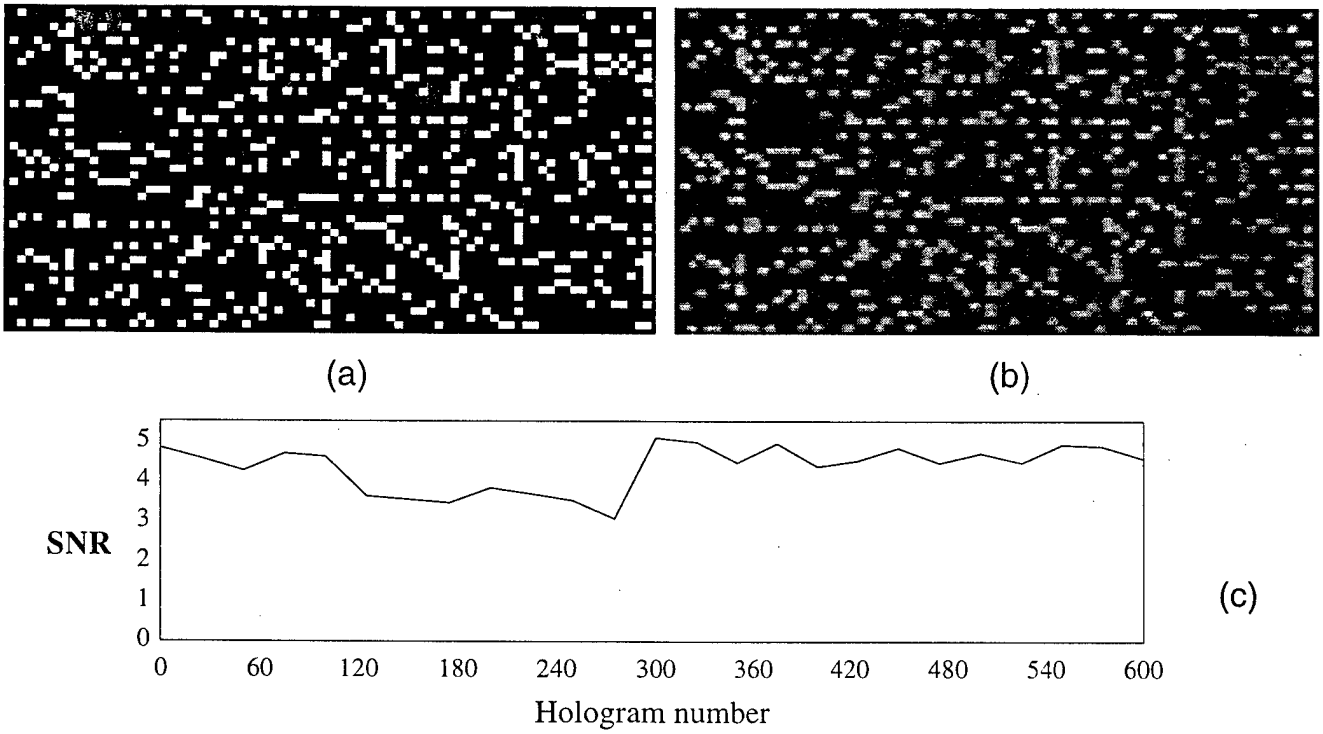


Figure 29. a) A binary page stored at location 450, b) the image recalled from location 450, and c) SNR for 600 data pages.

6.4. Data extraction

Data extraction consists of identifying the locations of data bits followed by a thresholding operation to determine the bit values. A simulator that we have developed as part of another project [BET97] provides a data extraction aid that aligns a grid over the data page to distinguish the boundaries between individual bits. It is often necessary to shift and resize this grid in order to throw out regions on the periphery of a page that suffer the highest noise effects due to lens aberrations and Gaussian illumination. Edge pixels are thrown out from each grid space and the remaining values are averaged to arrive at a single intensity value per data bit. It is also desirable to use a single threshold across a page for differentiating between these intensity values. As we have already mentioned, a hard threshold is only possible when intensity levels suffer only minor fluctuations across the page. A histogram is taken in order to identify the separation between the distributions of 1's and 0's. The threshold can be found from this histogram in a straightforward manner when the tails are non-overlapping. Otherwise, the threshold is calculated from statistical methods as in [GU96].

Sample reconstructions from our memory system are shown in Figures 30, 31, and 32. They are from an experiment in which 500 holograms of size 85×45 bits were recorded. The data blocks shown are of size 35×35 extracted from the center of each hologram. The pages were recorded in such a way that the grating strength increases as we get further into the schedule. Page 1, with relatively low power when recalled was early in the schedule and page 3 with a higher power was recorded towards the end. The series of figures shows the impact of inter-pixel cross–talk. On the average, each SLM pixel is sampled by 40 CCD pixels and a program produces an estimate for the intensity of an SLM pixel by averaging 18.5 CCD pixel values.

The segment of the original data mask, the image obtained by the CCD, the data page averaged from that image, and a histogram of the pixel intensities are shown in each figure. On the CCD image, the brightest pixels (light pixels) are identified with colors ranging from green to violet.

The data page images do not undergo any type of image enhancement, i.e. histogram equalization, contrast enhancement, matched filtering, etc., and we assume that a global threshold can be drawn that separates the light pixels from the dark pixels. Errors occur when the two distributions for light and dark pixels overlap.

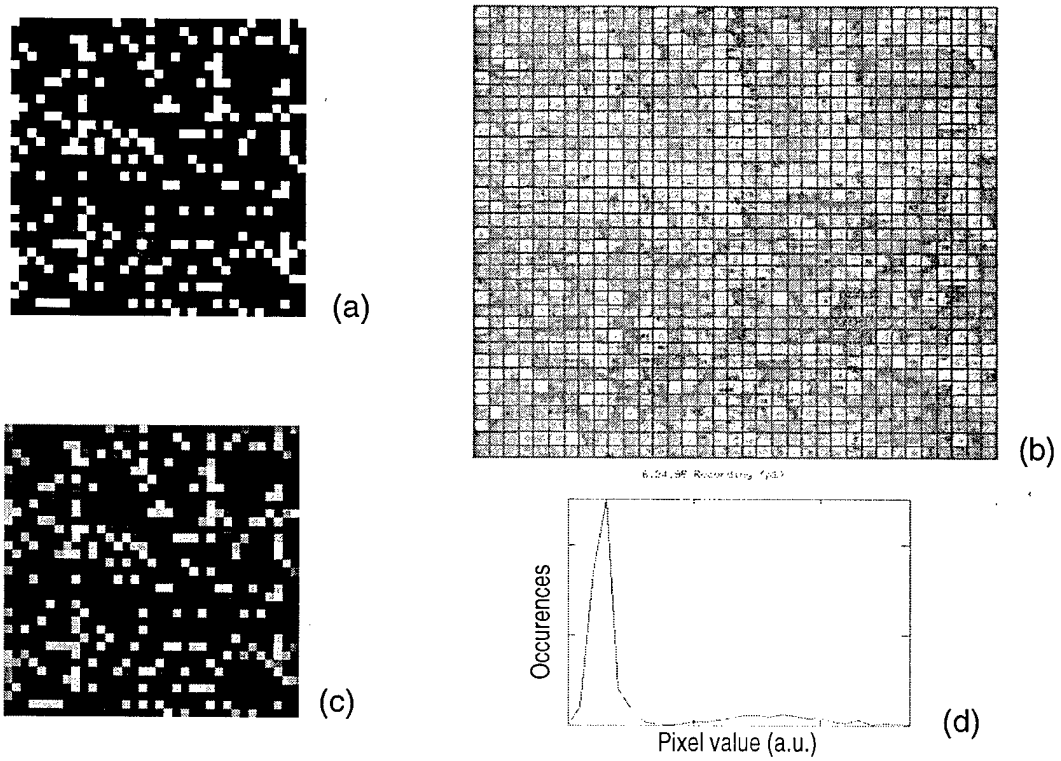


Figure 30. Page 1: (a) original data, (b) CCD reconstruction, (c) averaged, (d) intensity histogram.

Errors are identified in the figures by circles and pointed to by arrows. Figure 30 shows an image that corresponds to page 1 and exhibits no errors. Errors occur with both page 2 and page 3 which are recorded later in the schedule. Both reconstructions exhibit an increase in intensity. In examining the histograms, it is noted that with the increase in intensity there is a broadening of the distribution of light pixels which we will denote as DLIGHT, while the dark pixel distribution (DDARK) remains relatively tight. The errors that result can be attributed to the tail of DLIGHT spreading into DDARK. Figure 31 shows an image with two errors. The first, which is off center and to the left, is a dark pixel which is mistaken for being light. This is a situation where a dark pixel is surrounded by light pixels, so the most influential error is an intrapage interpixel cross–talk. This cross–talk can be mainly attributed to the finite size of the crystal which causes the "haloing" around single light pixels and clusters due to the loss of higher spatial frequencies. The reason why interpixel cross–talk increases while overall intensity increases is simply the result of the additional intensity that there is to spread. The error in the lower left corner is a light pixel categorized as a dark pixel. Since this occurs on the edge near the corner, it is likely to be attributed to the gaussian profile of the beam.

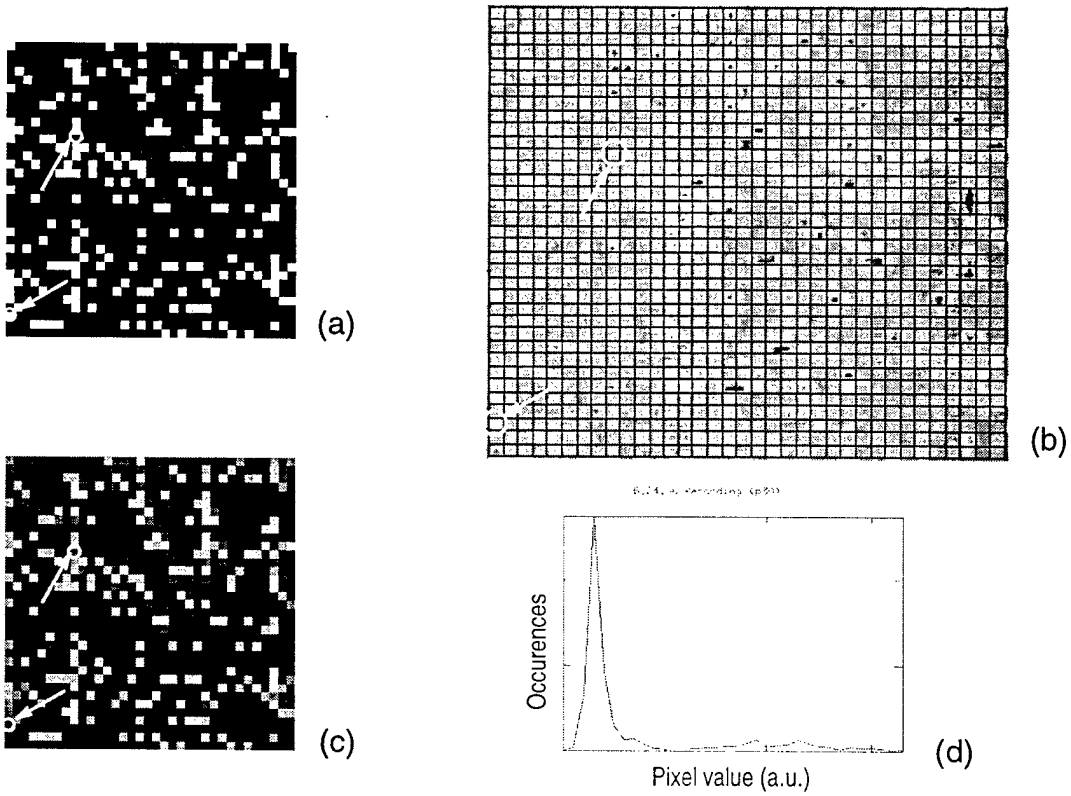


Figure 31. Page 2: (a) original data, (b) CCD reconstruction, (c) averaged, (d) intensity histogram.

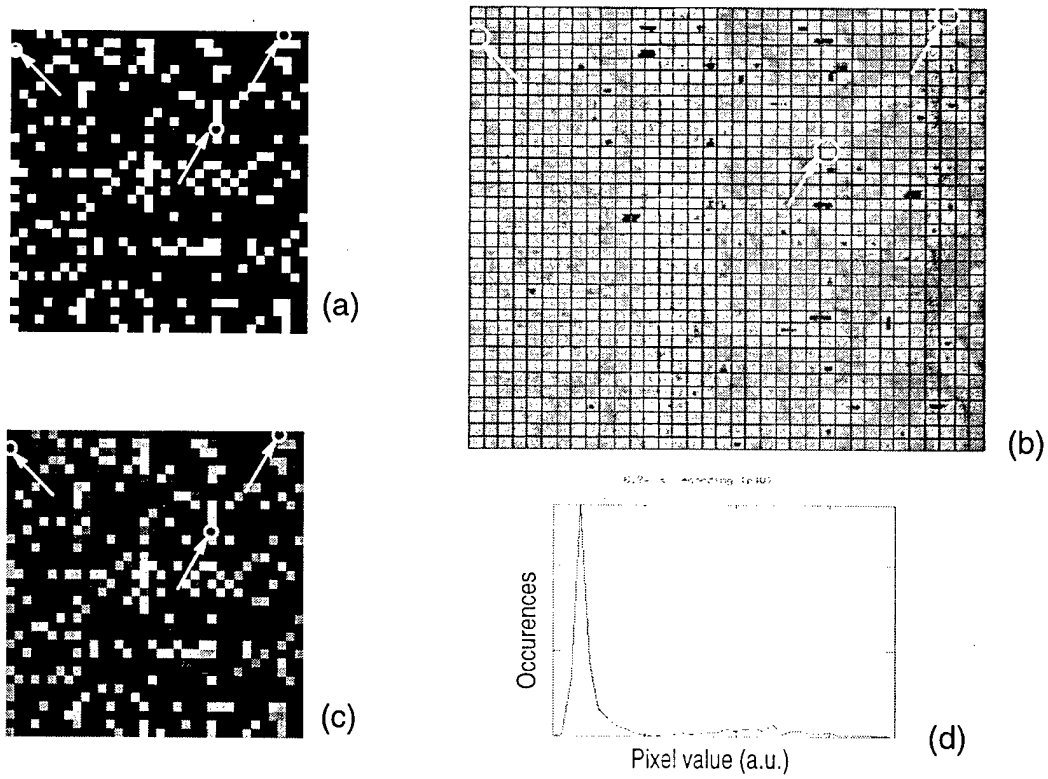


Figure 32. Page 3: (a) original data, (b) CCD reconstruction, (c) averaged, (d) intensity histogram.

Page 3, the image in Figure 32, shows errors that are of the same type. On the right hand side of the page there are occurrences of intrapage interpixel cross–talk where intensities of light pixels "bleed" into neighboring dark pixels. The error in the upper left corner is a light pixel that has lost intensity due to gaussian illumination.

Results from our system have shown errors to be most prominent on the edges of the data page. This situation is either handled by performing a regional thresholding or by throwing the data bits out entirely in the process described above. When removing these errors imposed by the aberrations and Gaussian illumination, a constant threshold methodology has been able to achieve a RBER below 10^{-3}.

6.4.1. Associative searching results

The behavior of VHDS-3 operating in the associative recall mode was similar to that of VHDS-2 and reinforced our observations with regards to the factors affecting the reconstruction of the reference beam plane. The improvement in the quality of the reference beam profile was significant, as can be seen in Figure 33. In this plot, the difference between the correlation peaks and the background noise level has noticeably increased and leads to a better detection of hits. However this difference is still less than what we observed when we stored pseudo-random data.

Using VHDS–3 we are experimenting to determine how the size of the bits within a page affects searching. Preliminary results have shown that we can indeed decrease the bit size and still search the memory effectively. A reduction in bit size translates directly to an increase in the storage capacity of the memory.

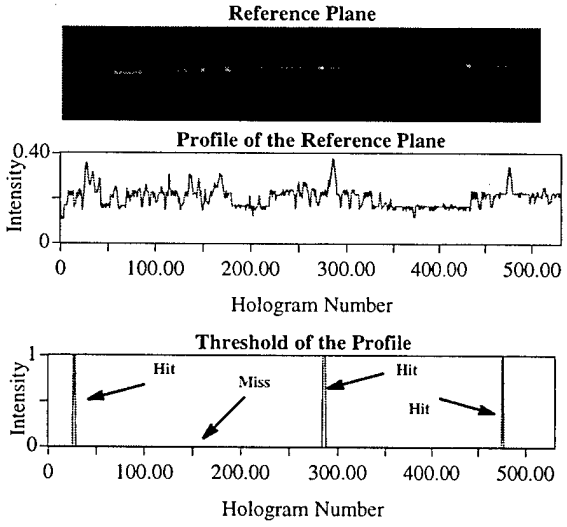


Figure 33. Results from a search for the data contained in pages 27, 154, 277, and 475.

6.5. Combinations of binary data and grey–scale images

In an effort to evaluate the ability of VHDS to accommodate different classes of data, including analog images, we performed a series of experiments that involved storing combinations of digital (binary) data pages and analog (grey scale) images to the memory. The purpose of including analog data is to determine if items such as faces or even video frames can be stored and efficiently searched using VHDS. This would make our system suitable for multimedia database management and allow searches with either analog or binary search arguments.

In one such experiment we stored 285 digital and 15 analog pages for a total of 300 holograms recorded at an angular separation of 0.0402°. Each digital page contained 210 characters comprising one record from a database whose fields are the last name, first name, company, address, city, zip code, telephone number, and e-mail address of individuals. The code blocks of each digital page were arranged in three rows of ten blocks for a total of 3600 bits per page. The analog pages contained various images at VGA resolution.

The experiment searched for both analog and digital data using entire as well as partial pages as search arguments. Using complete pages there were a total of 27 searches resulting in only one miss of the page that contained the lowest intensity analog data. Typical results are shown in Figure 34. Two observations can be readily made from the profiles in this figure: a) peaks are in general more distinct than those in the previous plots due mainly to a larger angular separation of holograms, and b) the average intensity and peak values for analog searches are much higher than those for digital searches. The reason for the second effect is the higher average intensity of analog pages. The reference beam profiles of analog searches contain also local maxima for almost every analog page when searching for an entire page because the analog images have similar features that correlate to the search argument better than the binary structure of a digital page.

We have also performed searches with different sizes of search arguments for both the digital and analog pages, and we found out that analog searches give very nice reference beam peaks for search arguments as small as 2/15ths of a page. Digital searches with such small arguments generated a lot of false hits and only when the search argument size was raised to 4/15ths of a page the number of false hits went down to 0–3 per search.

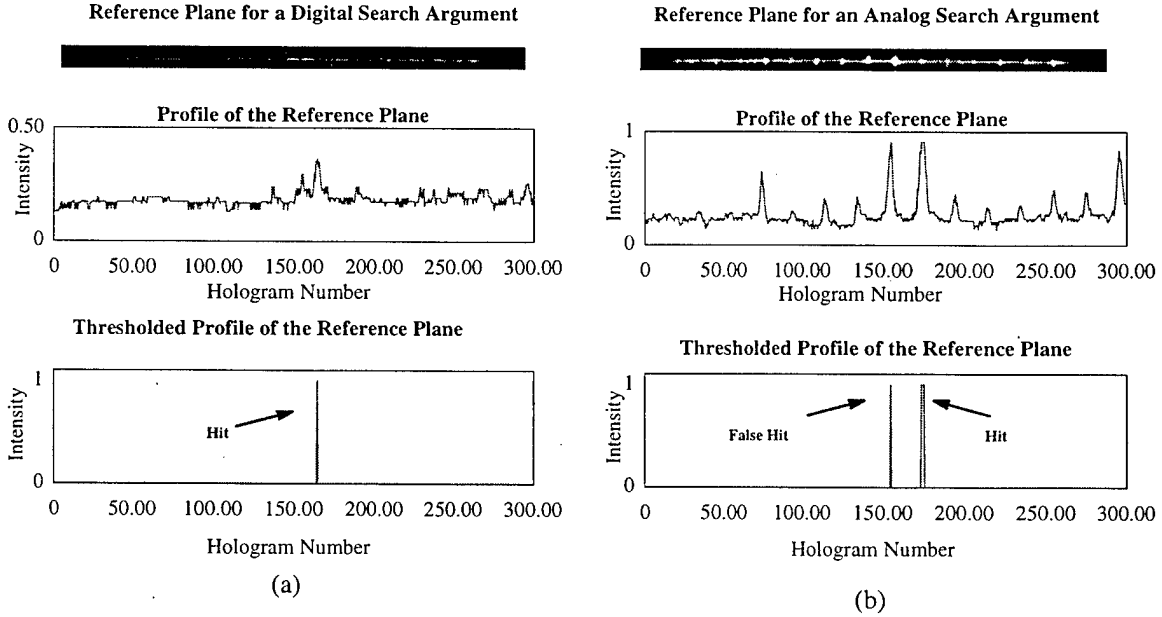


Figure 34. Associative recall of (a) a digital and (b) an analog data page.

Overall, this last round of experiments yielded better results in terms of the number of false hits and misses, as can be seen by comparing the match plots in Figures 35 and 36. The circles in these plots represent hits (including false hits) from a search. Each vertical line of circles is a search for one specific hologram. The positive sloped line in the plots is the mapping function

from pixel number to hologram number. Any of the circles that are exactly on this line or very close to it represent true hits, the rest correspond to false hits. The second match plot contains far fewer false hits indicating that mixing analog with digital data and further separating holograms in the crystal will produce better associative recalls.

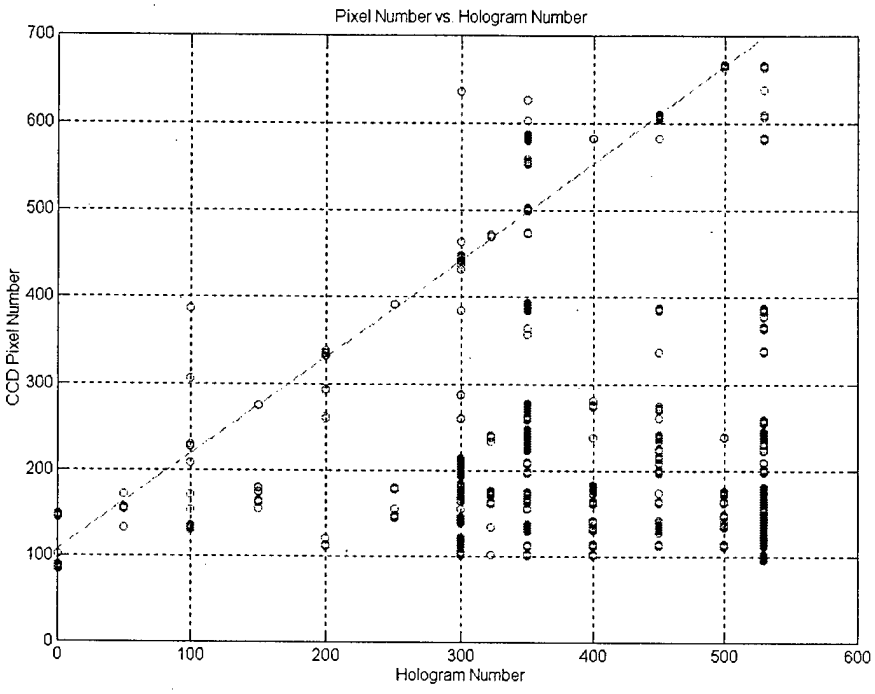


Figure 35. Match plot for VHDS–2 with 530 holograms of alphanumeric data.

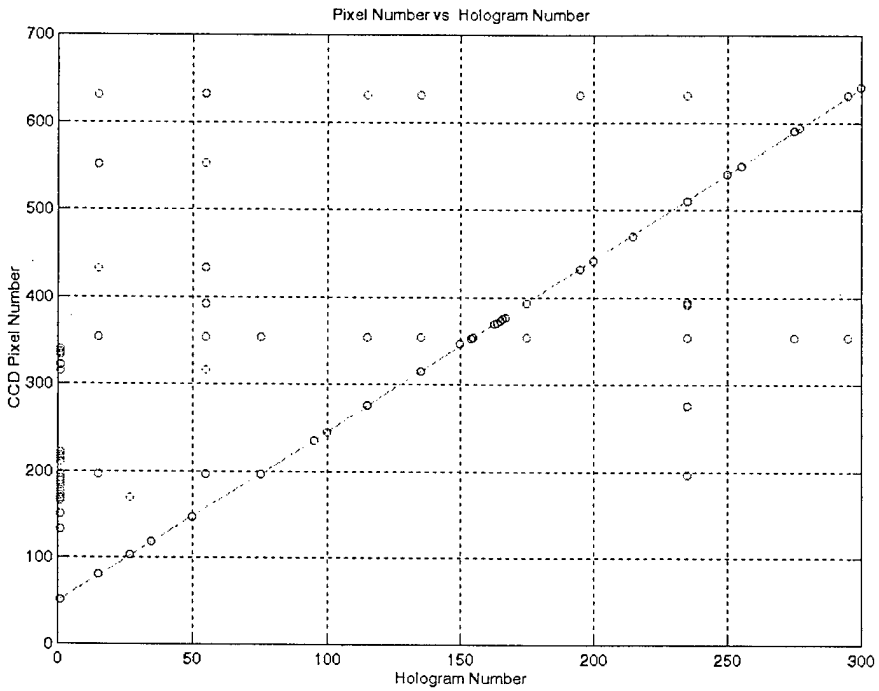


Figure 36. Match plot for VHDS–2 with 300 holograms of alphanumeric and image data.

6.6. Effect of data type on associative recall

We also compared a reference beam profile from VHDS-1 when random digital data were stored to reference beam profiles from VHDS-3, when real digital data and combinations of digital/analog data were stored. In summary, we observed that identification of hits in the reference beam profile becomes more difficult as we move from grey–scale images, to random binary, and to real binary data. This is because the contrast ratio, *CR*, defined as the ratio between the maximum peak intensity, I_{max}, over the average intensity, I_{av}, of the beam profile, decreases, as illustrated in the profiles of Figure 37. One would expect that the plots in b) and c) should be quite similar since in both cases digital data with 210 characters per page were stored. However, the hit peaks for the random data were more distinct than for the real data. We believe that the reason for this is the lower cross-correlation between the random data pages. When searching is performed, a correlation between the search argument and the stored pages occurs. The correlation is greatest when there is an exact match between the search argument and a data page, but the correlation also produces some output when the data and the search argument are similar. The output from similar pages increases the noise floor and makes finding actual hits difficult. For random data the degree of similarity drops significantly, and therefore, the noise level remains low and hits are easier to detect. This reasoning explains why hit peaks in b) are easier to see than the hit in c).

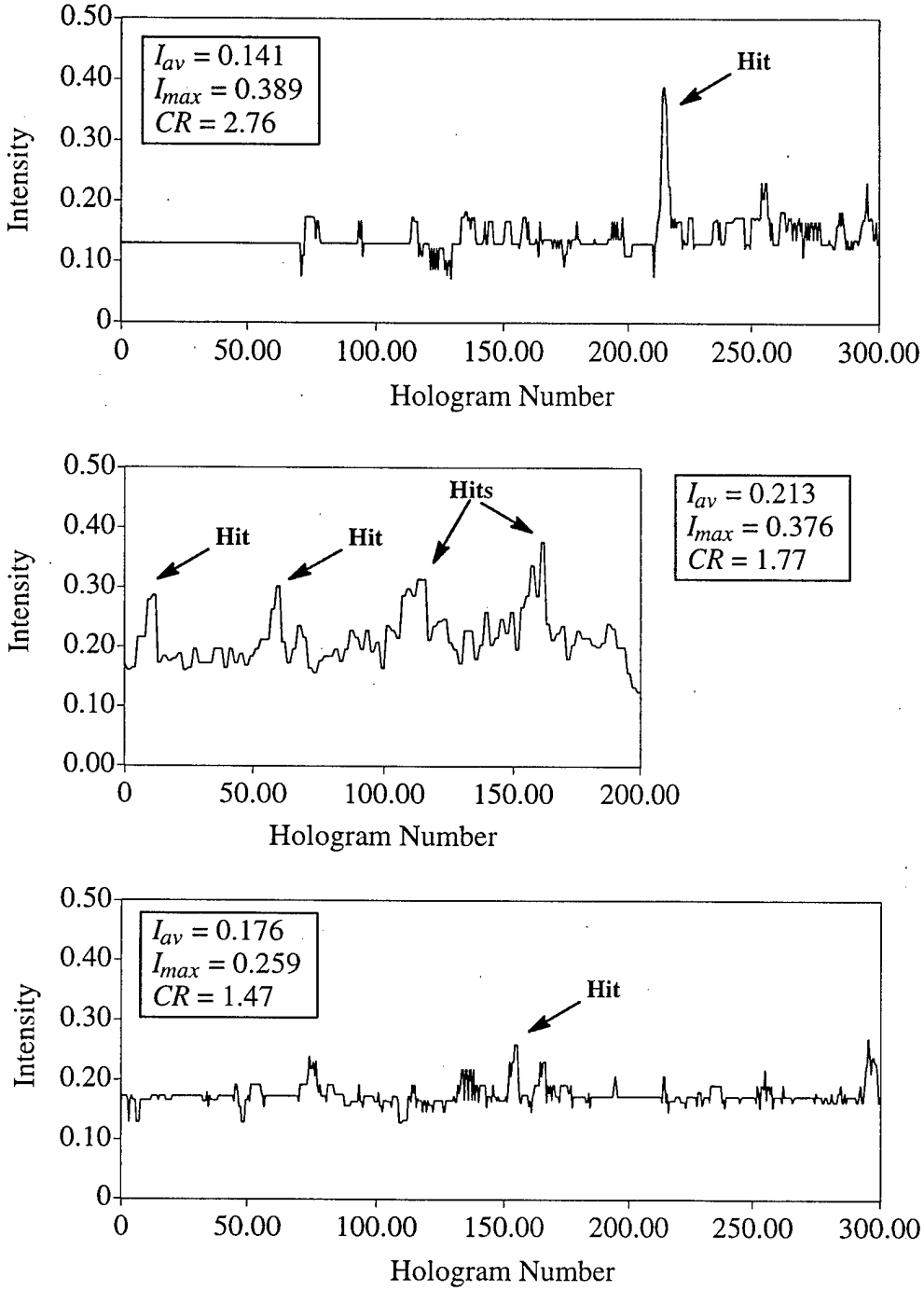


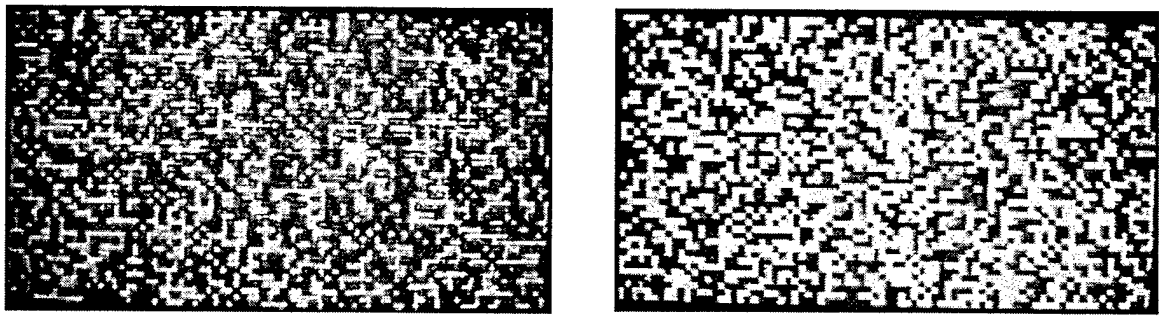
Figure 37. a) Associative recall with an analog search argument size of 16.7%. b) Associative recall of random binary data (search argument size is 33.3%). c) Associative recall for real binary data (search argument size is 26.7%).

6.7. System errors

During the course of this research we have observed several significant sources of error. The first and most obvious is that the intensity profile of the laser beam that is modulated by the SLM is Gaussian, which means that the intensity of the 'on' bits towards the center of the image are brighter than those at the edges. When the intensity of the 'on' bits is not uniform across a recovered data page, a hard thresholding function during data extraction cannot be used and other more tedious techniques must be employed.

The second type of error is due to blurring and has three major sources in our system. The first and least significante of the three is the finite size of optical elements that causes low pass filtering to occur as the modulated laser beam is propagated through the system. The second is that since we record Fourier transform holograms, the DC component of the data beam is brighter than high frequency components, and therefore, the high frequency terms are attenuated during recording. Finally, the Gaussian profile of the reference beam further aggravates this problem because it too has a higher intensity in its center. These factors are equivalent to a low pass filter that the data have to go through before they reach the output CCD.

A last source for error in these systems is misalignment due to component settling. Figure 38 shows two images, one taken after the components were allowed to settle for about a month and one captured after prototype three was realigned. Notice that the second image has much sharper edges and better uniformity. The combined effect of all noise sources in our system is significant, but we believe that each individual problem can be addressed and probably overcome in a commercially viable system.



(a) (b)

Figure 38. (a) Image captured after component settling, (b) image captured after a realignment.

As part of another research project, we have implemented a software package that can simulate any POM system and the noise sources in such a memory [BET97]. The simulator, called OASIS, has been used to examine first order errors, such as Gaussian illumination, in VHDS-2. When compared to actual data from VHDS-2, the simulator output showed that about 90% of the errors were introduced by those listed above. The other 10% occur due to second order noise effects that have yet to be implemented in the simulator.

7. Conclusions and Future Work

We have presented a volume holographic database system which is able to store and recall information both by address and associatively. The system can potentially accommodate databases of up to 10 million binary encoded records. The implementation includes an error-correcting code which allows industry standard bit error rates to be achieved. The code provides a simple decoding method and allows page rereads to decrease the bit error rate without significantly increasing the average response time. It must be noted here that decoding takes place only when the system operates in the addressed-based mode and not during associative recall.

We were able to record several hundred relational database records digitally encoded according to our coding scheme. We have demonstrated that the database can be accessed by address or by content. We focused our efforts towards improving and characterizing associative recall and have demonstrated successful recalls with search arguments as small as 1/30 of a data page. The results show that good associative addressability can be achieved by thresholding the reconstructed reference beam profiles. More elaborate signal processing techniques can be used to further reduce the percentage of false hits but they are outside the scope of this paper. We have also shown that such a memory provides position, orientation, and bit size variance which is desirable when searching relational databases. Both digital and analog data can be combined in the same recording and searching can be performed on both types of data. This functionality can be very useful in multimedia database management systems and a whole array of applications that use different types of data. In performing experiments that combined both analog and digital data we were able to observe that the intensity of hits in the reference plane is dependent not only upon search argument size, but also content or the stored data's diffraction efficiency. This means that steps need to be taken to ensure that both the digital and analog data pages are recorded at the same efficiency. Mixing alphanumeric and analog data provided these benefits: a) wider angular separation, b) greater intensity of reconstructed reference beams when searching for analog data, c) lower cross–correlation due to mixed data types and fewer binary data pages, and d) the ability to remove some false hits that corresponded to analog page positions when digital search arguments were used and vice versa.

Currently only recording is fully automated. For VHDS to become a realistic system, every operation must be automated and that includes recall and searching. Automatic address-based recall will require the development of software to capture images from the cameras and convert them into database records of the format that the electronic host can recognize. This process also entails the development of mechanisms for signal recovery, demodulation, and error detection and correction. Automation of associative recall will require the development of software to control both the SLM and the cameras simultaneously, and the implementation, in hardware or software, of algorithms that can successfully detect hits in the reference plane and convert them into address locations. With the addition of a suitable feedback mechanism to retrieve qualifying pages, our associative memory system will be completely automated.

Other future work should include memory improvements and more specifically attempts to increase the memory's capacity, to reduce the BER, and to improve the contrast between hits and non-hits in the reference plane. Capacity increase can be accomplished by improving the optics' quality to enhance the system's resolution, which may also lead to a lower BER. Capacity can also increase through the use of smaller bit size on the SLM, which allows more bits per page,

and by using all available addresses. Finally, capacity can be increased through the use of spatial multiplexing, but this may slow down associative searching since each spatial location will need to be searched serially.

Improving contrast between hits and non-hits in the reference plane involves increasing the orthogonality between encoded data pages. This means that we need to develop a modulation scheme that produces random looking data pages. Also, if every page has the same number of ON bits, then finding hits may become simpler because variations in diffraction efficiency between pages will be reduced. This includes analog data, which currently exhibit the largest diversity in diffraction efficiency. A possible solution would be the adoption of dual rail encoding, or storing both the image and its complement. Future work on this project should also include performing a study of the effects of the encoding scheme on the quality of the associative recall. More accurate statistical analysis and bit error study of the output channel will lead to improvements to the coding scheme.

We also plan to replace the liquid crystal display which absorbs a lot of power with a TI digital micromirror display (DMD). An additional objective would be to attempt to eliminate the moving parts (translation stage in the reference beam arm) of VHDS by using a second SLM (probably another TI–DMD device) to generate the reference beam. Since the DMD is a highly reflective device we may use clusters of pixels to deflect the reference beam at different angles.

7.1. Multimedia Applications for VHDS

In this last section we briefly outline some possible extensions of VHDS and show initial results of a video indexing experiment. VHDS can be configured as a special–purpose opto-electronic processing unit that can be added as a peripheral to an electronic host and undertake computationally intensive operations (mainly content–based searches) in a multimedia environment. In order to satisfy this objective we will have to determine which applications are better suited for our system by characterizing the response of VHDS for different types of data. We have chosen four applications each of which uses a different data type:

a) Database management (digital alphanumeric data)
b) Video indexing (gray scale and color images)
c) Image compression by vector quantization (binary images)
d) Interactive video (combinations of alphanumeric data, binary and gray scale images).

Each one of the selected applications requires searching a large database for the presence of a search pattern. There may be zero, one, or multiple hits. Electronic implementations are usually based on complex search algorithms, elaborate and memory consuming indexing, or content-addressable memories and processors. The latter come in small sizes (maximum size today is 16KB [MUS96]) and at high cost, thus their use is limited.

7.1.1. Video Indexing

Video indexing is the process of identifying the range of video frames in which a certain pattern (i.e., object, target, face, or signature) appears. It has many potential applications including military, scientific, and consumer ones. Due to the multitude of parameters involved and the amount of data that must be processed, video indexing is an extremely computationally intensive operation and, therefore, seldom available to the end user. The lack of standard techniques further complicates the problem.

When analog images are used, an optical correlator may provide an acceptable solution. Alternatively, the video sequence can be stored in a holographic memory and the search engine may be used to index it. When presented with the search pattern, the system will reconstruct the reference beam profile which should yield the qualifying areas. It must be noted here that it is not necessary to store the entire video sequence in the memory. Rather, a sampling frame rate can be adopted for the desired degree of accuracy. Figure 39 illustrates an example of video indexing.

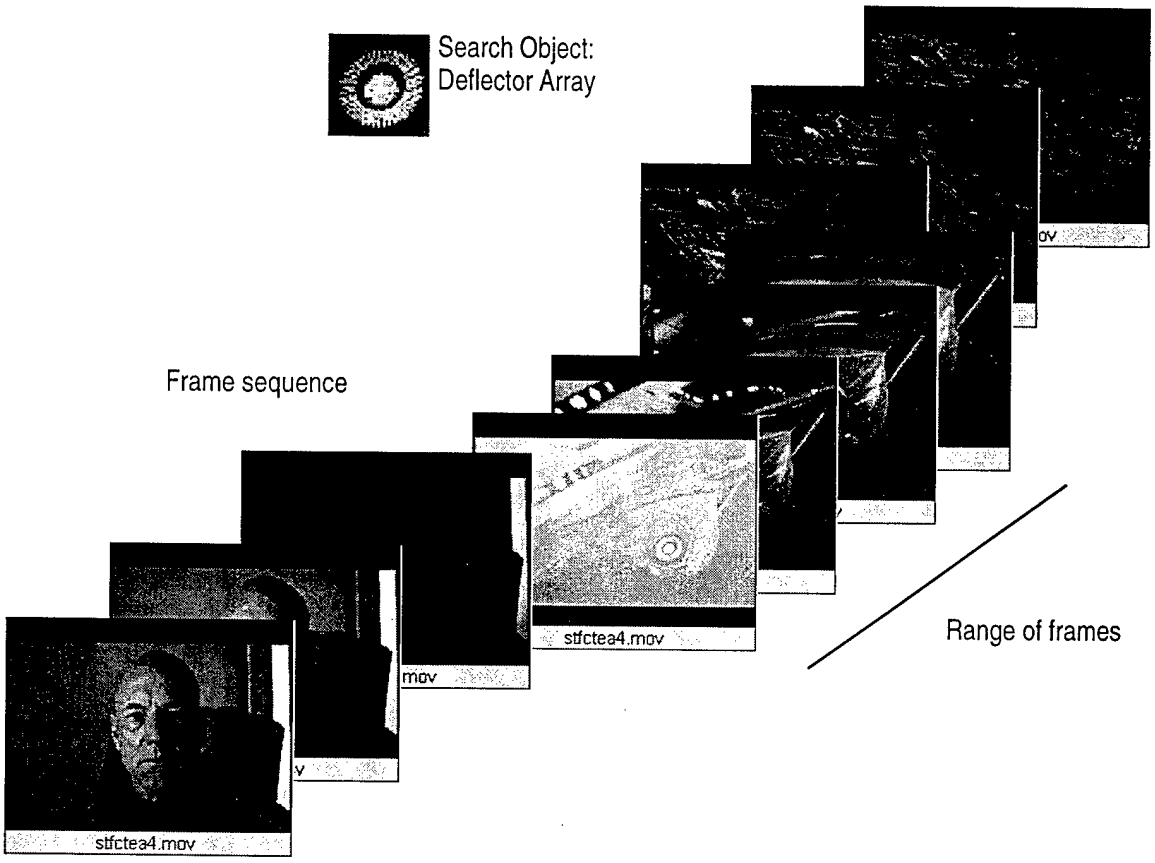


Figure 39. A video sequence from the "Star Trek: First Contact" feature movie. The indexing here is performed using the deflector array of the U.S.S. Enterprise.

In a recent experiment, we recorded a similar video sequence comprising 126 frames from the trailer for the last Star Trek movie. Sample video frames are shown in Figure 40 which includes a subsequence (frames in the middle row) containing a feature that looks like an explosion. When we used one of these frames as a search argument in VHDS, we generated the reference beam profile shown in Figure 41, which demonstrate that video indexing is indeed possible. Further experiments and analysis are, of course, necessary.

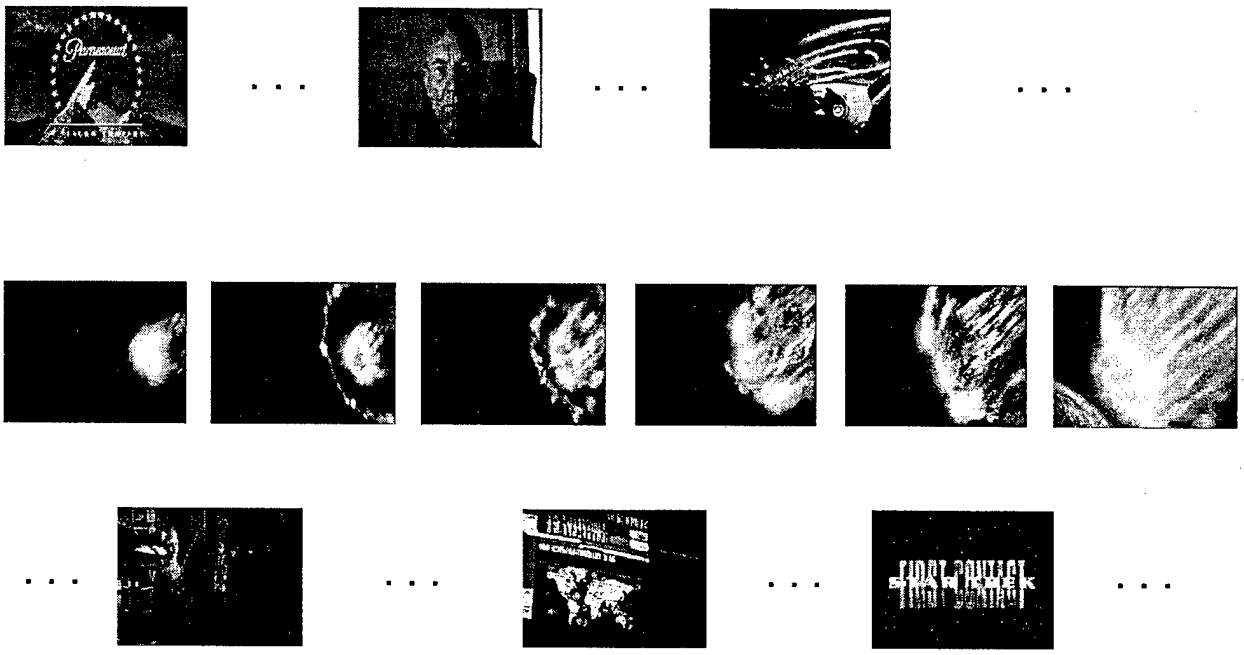


Figure 40. A sample of a 126 video frame sequence recorded in VHDS.

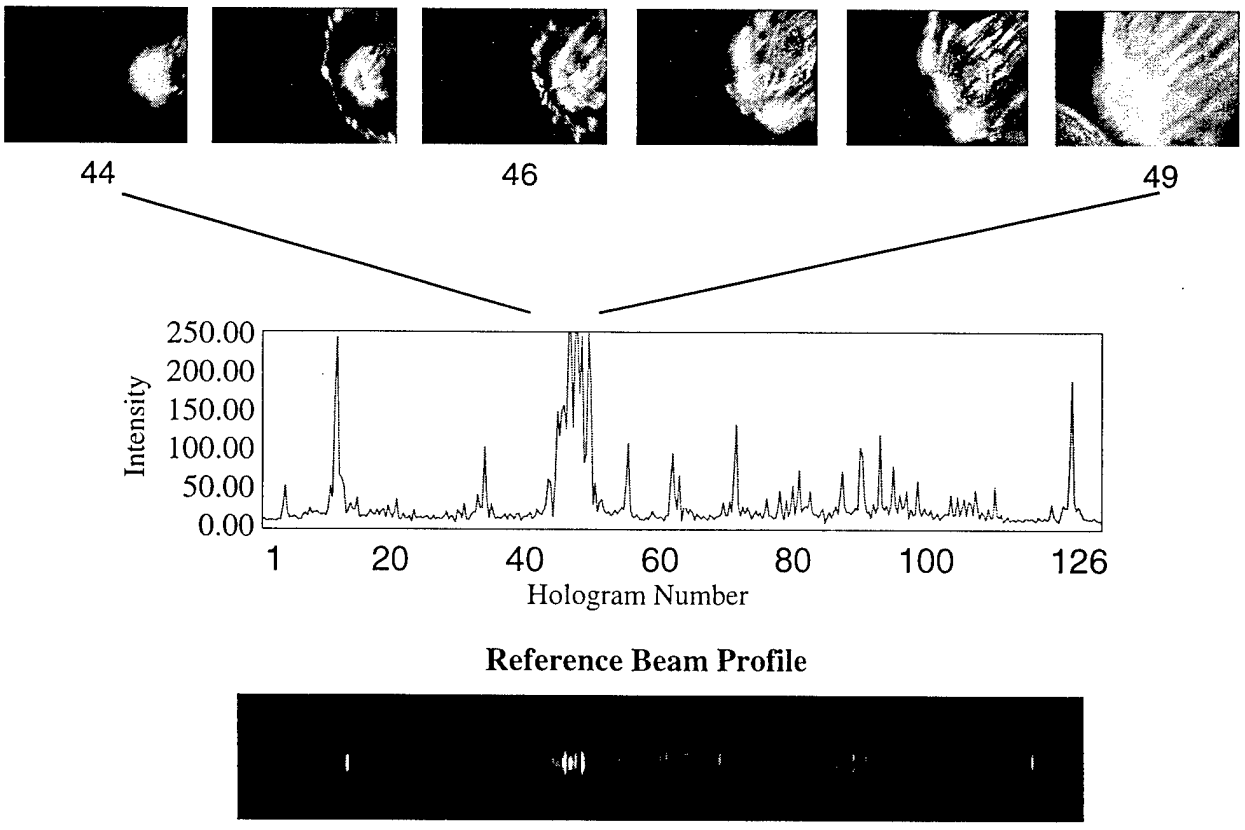


Figure 41. Results of video indexing experiment.

7.1.2. Interactive Video

Interactive video applications have to combine alphanumeric digital data with video frames. The user has the ability to stop the video sequence and access the so called "hypervideo" information related to the current frame. The search engine must be capable of handling both analog and digital data since the user may request access based on image content or a keyword.

Video frames can be stored in a holographic memory interleaved with pages containing the "hypervideo" information, as shown in the example of Figure 42. An associative search can be performed using either analog or digital arguments and the system can be designed to retrieve either video frames or the corresponding digital data. The holographic search engine can be used for several other applications that combine image and textual data, such as the management of medical databases, personnel files, and police records.

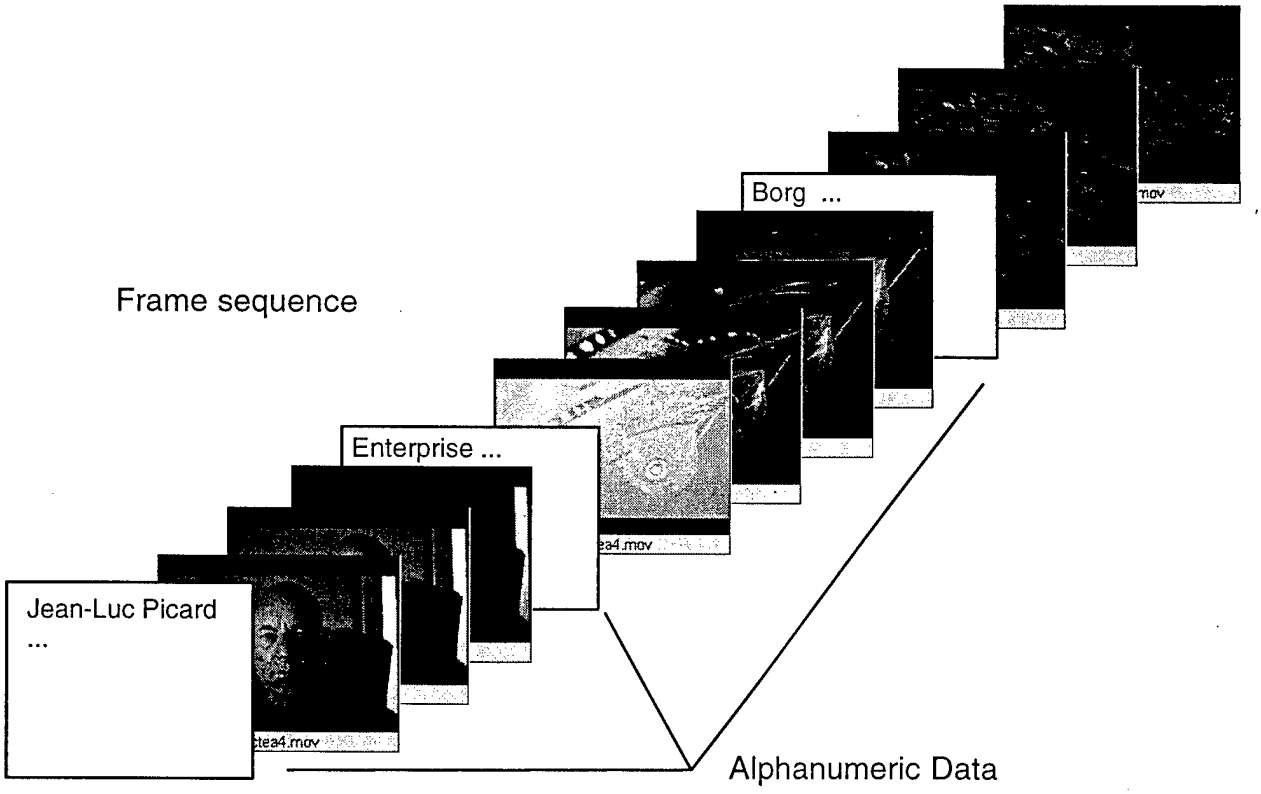


Figure 42. Example of data pages in an interactive video experiment.

7.1.3. Image Compression by Vector Quantization

Vector quantization is a lossy image compression technique, in which a digital image is first decomposed into n-dimensional image vectors which are subsequently compared to a set of codevectors, the codebook, to find the best match [GER92]. Compression is achieved by using a relatively small codebook compared to the possible image vectors. For large codebooks, the search process becomes computationally intensive. Elaborate software techniques and VLSI implementations compromise compression and quality to achieve fast processing.

An optical memory can potentially increase the efficiency of this image compression technique both in terms of speed and quality. If a sufficiently large codebook is stored in pages in the memory, then an image vector can be compared against the whole codebook in a single step. The codebook vector that best matches the image vector produces the reference beam with the highest intensity at the output and then this beam can be used to retrieve the index of the code-vector stored in the same page in the memory.

8. References

[BER89] P. B. Berra, A. Ghafoor, P. A. Mitkas, S. J. Marcinkowski, and M. Guizani, "The impact of optics on data and knowledge base systems," IEEE Trans. on Knowledge and Data Eng., Vol. 1, No. 1, March 1989, pp. 111–132.

[BET97] G. Betzos, M. Porter, J. Hutton, and P. A. Mitkas, "OASIS: An Interactive Tool for the Analysis of Page-oriented Optical Memories," submitted to Applied Optics, 1997.

[BUR94] G. W. Burr, F. H. Mok, and D. Psaltis, "Large scale volume holographic storage in the long interaction length architecture," in Photonics for Processors, Neural Networks, and Memories II, (ed. by J. L. Horner, B. Javidi, and S. T. Kowel), Proc. Soc. Photo-Opt. Instrum. Eng., Vol. 2297, 1994, pp. 402–414.

[BUR95a] G. W. Burr, F. H. Mok, and D. Psaltis, "Angle and space multiplexed holographic storage using the 90° geometry," Opt. Comm., Vol. 117, May 1995, pp. 49–55.

[BUR95b] G. W. Burr and D. Psaltis, "Improving diffraction efficiency of holographic memories via $LiNbO_3$:Fe oxidation state" presented in the Symposium on Holographic Storage Technology at the annual meeting of the Optical Society of America, Portland, OR, Sept. 1995.

[COL71] R. J. Collier, C. B. Burkhardt, and L. H. Lin, Optical Holography, Academic Press, New York, 1971.

[DEP97] Proceedings of the 1996 Workshop on Data Encoding for Page-oriented Optical Memories (DEPOM '96), P. A. Mitkas, Editor, Colorado State University College of Engineering, 1997.

[FAR79] P. G. Farrell, "Array Codes," in Algebraic Coding Theory, (ed. by G. Longo), Springer-Verlag, 1979, pp. 231–242.

[FAR82] P. G. Farrell and S. J. Hopkins, "Burst-error-correcting array codes," The Radio and Electron. Engineer, Vol. 52, No. 4, April 1982, pp. 188–192.

[FAR90] P. G. Farrell, "Coding as a cure for communication calamities: the successes and failures of error control," Electron. and Commun. Eng. J., Vol. 2, No. 6, Dec. 1990, pp. 213–220.

[FAR92] P. G. Farrell, "A survey of array error control codes," European Trans. on Telecommun., Vol. 3, No. 5, Sept.-Oct. 1992, pp. 441–454.

[GAB48] D. Gabor, "A new microscopic principle," Nature, Vol. 161, 1948, p. 777.

[GER92] A. Gersho and R. Gray, *Vector quantization and signal compression*, Kluwer Academic Publishers, 1992.

[GOE95a] B. J. Goertzen and P. A. Mitkas, "An Error Correcting Code for Volume Holographic Storage of a Relational Database," *Optics Letters*, Vol. 20, No. 15, Aug. 1, 1995, pp. 1655-1657.

[GOE95b] B. J. Goertzen, Volume Holographic Storage for Large Relational Databases, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, 1995.

[GOE96a] B. J. Goertzen, K. G. Richling, and P. A. Mitkas, "Implementation of a Volume Holographic Database System," presented at the International Topical Meeting on Optical Computing, OC '96, Apr. 21-25, 1996, Sendai, Japan.

[GOE96b] B. J. Goertzen and P. A. Mitkas, "Volume Holographic Storage for large Relational Databases," *Optical Engineering*, Vol. 35, No. 7, July 1996.

[GU96] C. Gu, F. Dai, and J. Hong, "Statistics of Both Optical and Electrical Noise in Digital Volume Holographic Data Storage," *Electronics Letters*, Vol. 32, No. 15, 1996.

[HAM86] R. W. Hamming, Coding and Information Theory, Prentice-Hall, Englewood Cliffs, NJ, 1986.

[HEA94] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Volume holographic storage and retrieval of digital data," Science, Vol. 265, No. 5173, August 1994, pp. 749–752.

[HON95] J. H. Hong, I. McMichael, T. Y. Chang, W. Christian, and E. G. Paek, "Volume holographic memory systems: techniques and architectures," Optical Engineering, Vol. 34, No. 8, Aug. 1995, pp. 2193–2203.

[IMA90] H. Imai, Ed., Essentials of Error-Control Coding Techniques, Academic Press, San Diego, 1990.

[IRA95] L. J. Irakliotis, S. A. Feld, F. Beyette, P. A. Mitkas, and C. W. Wilmsen, "Optoelectronic parallel processing with surface-emitting lasers and free-space interconnects," J. of Lightwave Technol., Vol. 13, No. 6, June 1995, pp. 1074–1084.

[LI94] H. S. Li and D. Psaltis, "Three-dimensional holographic disks," Appl. Opt., Vol. 33, No. 17, June 1994, pp. 3764–3774.

[MAB91] A. O. Mabogunje and P. G. Farrell, "Burst Error Correction Capability of Square Array Codes," Electron. Lett., Vol. 27, No. 13, June 1991, pp. 1215–1216.

[MAN91] E. S. Maniloff and K. M. Johnson, "Maximized photorefractive holographic storage," J. Appl. Phys., Vol. 70, No. 9, Nov. 1, 1991, pp. 4702–4707.

[MOE95] W. E. Moerner et. al., "Photorefractive polymers for holographic optical storage," presented in the Symposium on Data Storage at the annual meeting of the Optical Society of America, Portland, OR, Sept. 1995.

[MAR90] A. B. Marchant, Optical Recording, Addison-Wesley, Reading, MA, 1990.

[MIT94] P. A. Mitkas and L. J. Irakliotis, "Three–dimensional Optical Storage for Database Processing," invited paper, J. of Opt. Mem. and Neural Networks Vol. 3, No. 2, 1994, pp. 217–229.

[MOK93] F. H. Mok, "Angle-multiplexed storage of 5000 holograms in lithium niobate," Opt. Lett. Vol. 18, No. 11, June 1, 1993, pp. 915–917.

[MUS96] MUSIC Semiconductors, *MU9C2480A LANCAM*, Colorado Springs, CO, 1996.

[NEI93] M. A. Neifeld, "Computer generated holography for optical memory using sparse data words: capacity and error tolerance," Applied Optics, Vol. 32, No. 26, Sept. 1993, pp. 5125–5134.

[NEI94a] M. A. Neifeld and J. D. Hayes, "Parallel Error Correction for Optical Memories," J. of Opt. Mem. and Neural Networks, Vol. 3, No. 2, Feb. 1994, pp. 87–98.

[NEI94b] M. A. Neifeld and M. McDonald, "Error correction for increasing the usable capacity of photorefractive memories," Opt. Lett., Vol. 19, No. 18, Sept. 1994, pp. 1483–1485.

[NOR94] G. P. Nordin and P. Asthana, "Achieving a minimum signal-to-noise ratio in angularly multiplexed volume holographic optical data storage systems," in Photonics for Processors, Neural Networks, and Memories II, (ed. by J. L. Horner, B. Javidi, and S. T. Kowel), Proc. Soc. Photo-Opt. Instrum. Eng., Vol. 2297, 1994, pp. 392–401.

[ONE93] P. E. O'Neil, "The set query benchmark," in The Benchmark Handbook for Database and Transaction Processing Systems, (ed. by J. Gray), Morgan-Kaufman, San Mateo, CA, 1993, pp. 359–396.

[OWE89] Y. Owechko, "Nonlinear holographic associative memories," IEEE J. of Quantum Electron., Vol. 25, No. 3, March 1989, pp. 619–634.

[PLE89] V. Pless, Introduction to the Theory of Error-Correcting Codes, John Wiley & Sons, Inc., New York, 1989.

[POR98] M. Porter, Data Recovery Schemes for Page–oriented Optical Memories, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, to be defended in 1998.

[PSA95a] D. Psaltis, "Holographic memories," Scientific American, Vol. 273, No. 5, Nov. 1995, pp. 70–76.

[PSA95b] D. Psaltis, "Large scale holographic memories," presented in the Symposium on Coding and Signal Processing for Highly Parallel Optical Storage Systems at the annual meeting of the Optical Society of America, Portland, OR, Sept. 1995.

[RED92] S. Redfield, "Holographic storage: not a device but a storage class," in Enabling Technologies for High-Bandwidth Applications, Proc. Soc. Photo-Opt. Instrum. Eng., Vol. 1785, 1992, pp. 45–51.

[RIC96] K. G. Richling and P. A. Mitkas, "Characterization of Volume Holographic Associative Searching," presented at the OSA Holography Topical Meeting, Apr. 28-30, 1996, Boston, MA.

[RIC97a] K. G. Richling, Characterization of Associative Data Recall in Volume Holographic Memories, M.Sc. Thesis, Dept. of Electrical Engineering, Colorado State University, 1997.

[RIC97b] K. G. Richling, B. J. Goertzen, and P. A. Mitkas, "Holographic Associative Processing for Large Databases," to be published, Applied Optics, 1997.

[RIC97c] K. G. Richling, P. A. Mitkas, S. Mailis, L. Boutsikaris, and N. Vainos, "Holographic Storage and Associative Processing of Analog and Digital Data," Proceedings of the IEEE LEOS 10th Annual Meeting, San Fransisco, CA, November 10–13, 1997, Vol. 1, pp. 132-133.

[SHE95] R. M. Shelby et. al., "Defining materials limits to holographic data storage: Holographic storage tester," presented in the Symposium on Digital holographic Data Storage: Materials and Devices at the annual meeting of the Optical Society of America, Portland, OR, Sept. 1995.

[WUL94] J. R. Wullert II and Y. Lu, "Limits of the capacity and density of holograhic storage," Appl. Opt., Vol. 33, No. 11, April 1994, pp. 2192–2196.